

DLR-IB-RM-OP-2018-248

**Analysis and Comparison of
Cooperative Control Architectures
for Landing a Small Fixed-Wing
Aircraft on a Mobile Platform**

Bachelor's Thesis

Wilson José de Sá Marques



DLR

**Deutsches Zentrum
für Luft- und Raumfahrt**

BACHELORARBEIT

ANALYSIS AND COMPARISON OF COOPERATIVE CONTROL ARCHITECTURES FOR LANDING A SMALL FIXED-WING AIRCRAFT ON A MOBILE PLATFORM

Freigabe:

Der Bearbeiter:

Unterschriften

Wilson José de Sá Marques



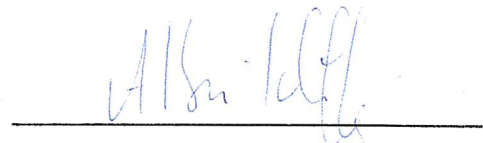
Betreuer:

Tin Muskardin



Der Institutsdirektor

Dr. Alin Albu-Schäffer



Dieser Bericht enthält 112 Seiten, 71 Abbildungen und 17 Tabellen

Wilson José de Sá Marques

**Analysis and Comparison of Cooperative
Control Architectures applied in Landing a
Small Fixed-Wing Aircraft on a Mobile
Platform**

Campos dos Goytacazes – Brazil

2018

Wilson José de Sá Marques

**Analysis and Comparison of Cooperative Control
Architectures applied in Landing a Small Fixed-Wing
Aircraft on a Mobile Platform**

Thesis presented to the Instituto Federal Fluminense, Brazil, as a partial requisite to the awarding of the five-year Bachelor degree in Control and Automation Engineering.

Instituto Federal de Educação, Ciência e Tecnologia Fluminense - IFF Campos – Centro
Control and Automation Engineering Program

Supervisor: Dr. Alexandre Carvalho Leite

Co-supervisor: Tin Muskardin (German Aerospace Center – DLR)

Campos dos Goytacazes – Brazil

2018

Acknowledgements

To God for life.

To my professor Alexandre Carvalho Leite for giving me the opportunity to work in such an amazing topic, and for have always directed me towards the right path. You contributed a lot to my professional and personal development.

To my supervisor in Germany, Tin Muskardin, a special thanks for all his dedication to this work. For the time spent in discussions (in person and over skype) to answer many questions that have arisen during this work. You are an example of professional and a spectacular human being, always solicitus and willing to teach and help. I learned a great deal from you.

To Dr. Konstantin Kondak for accepting me as a peer in the Flying Robots Group and to all the other friends at DLR for the great time we spent together working, and during conversations at coffee breaks and lunch time, specially to Georg Balmer, Sven Schmid, Juanito and JP.

To my friend Benício for all the discussions and advices that much contributed to this work. Besides all the conversations always so interesting and funny. I admire a lot your way of seeing our world and I'm always learning something new from you.

To my friend André for all the help during my time in Germany. For all the encouragement at work, for the help with the German language and for all the experiences and funny moments that we shared. I was lucky for having you there with me.

To my parents, Cristina and Alexandre for the love, comprehension and encouragement during my whole life.

Abstract

The successful landing of a fixed-wing Unmanned Aerial Vehicle (UAV) on top of an Unmanned Ground Vehicle (UGV) has been accomplished by researchers at the German Aerospace Center (DLR). Still, some aspects of the cooperative controller require further analysis and there is some room for improvement in the control strategy, most importantly, the robustness of the system needs to be assessed. In this thesis, modifications within the current cooperative control architecture are proposed in order to make the system faster and more robust. Different PID-based controller structures are used to close the loop and connect the vehicles. An optimization-based method is employed to select an optimal set of controller parameters. Finally, a sensitivity analysis test is performed using two approaches: a time-domain analysis, based on Monte Carlo simulations, and a frequency-domain analysis.

Keywords: UAV, UGV, cooperative control, robustness, optimization, sensitivity analysis.

*“My mind rebels at stagnation. Give me problems, give me work,
give me the most abstruse cryptogram or the most intricate analysis,
and I am in my own proper atmosphere.”
(Sherlock Holmes)*

Contents

	List of Figures	11
	List of Tables	15
1	INTRODUCTION	21
1.1	Application context	21
1.2	Goal and objectives	22
1.3	A brief project review	22
1.3.1	The cooperative landing maneuver	22
1.3.2	Cooperative Control	24
1.3.3	UAV platform overview: Penguin BE	25
1.3.4	UGV landing platform overview: a modified passengers car	25
1.4	Outline of thesis	26
2	VEHICLE MODELS REVIEW	27
2.1	UAV Model	27
2.2	UGV Model	28
2.3	Linearization	29
2.3.1	Linear aircraft dynamics	29
2.3.1.1	Longitudinal model	30
2.3.1.2	Lateral model	31
2.3.2	Ground Vehicle	31
2.3.2.1	Cruise	32
2.3.2.2	Steering	32
2.4	Stability	33
2.4.1	Aircraft Dynamic Modes	34
2.4.2	Car Poles	34
3	COOPERATIVE CONTROL ARCHITECTURES	36
3.1	Current Architecture	36
3.2	Proposed Architecture	36
3.3	SISO and MIMO Systems	37
3.4	State-Space Representation	37
3.5	Position Alignment and Touchdown	38
3.5.1	Longitudinal Control	39
3.5.1.1	Current Architecture	39

3.5.1.2	Proposed Architecture	39
3.5.2	Lateral Control	40
3.5.2.1	Current Architecture	40
3.5.2.2	Proposed Architecture	40
3.5.3	Vertical Control	42
3.5.3.1	Landing Requirements	42
3.5.3.2	Flare Law	43
3.5.3.3	Current Architecture	44
3.6	Finite State Machine	44
4	COOPERATIVE CONTROL DESIGN	47
4.1	Control Loops	47
4.2	Disturbances and Uncertainties	48
4.2.1	Wind Gust	48
4.2.2	Wind Turbulence	49
4.2.3	Time Delays	50
4.3	Frequency Domain Design	50
4.4	PID Controller	55
4.4.1	Controller Tuning	55
4.5	Stability Assessment	58
4.6	Control Validation	61
4.6.1	Simulation Tests	61
4.6.2	Landing	62
5	SENSITIVITY ANALYSIS	67
5.1	Definition	67
5.2	Sensitivity in the Time Domain	67
5.2.1	Monte Carlo Simulation	67
5.3	Sensitivity in the Frequency Domain	71
5.3.1	Feedback Fundamentals	71
5.3.2	Design Considerations	73
6	RESULTS AND DISCUSSION	75
7	CONCLUSION AND OUTLOOK	84
	BIBLIOGRAPHY	85

	APPENDIX	88
	APPENDIX A – INDIVIDUAL VEHICLE CONTROL DESIGN	89
A.1	Flight Control	89
A.1.1	Successive Loop Closure	89
A.1.2	Lateral Control	90
A.1.2.1	Lateral Stability Augmentation and Autopilot	91
A.1.3	Longitudinal Control	92
A.1.3.1	Longitudinal Stability Augmentation and Autopilot	92
A.1.4	TECS	93
A.2	Ground vehicle control	94
A.2.1	Longitudinal Control	94
A.2.2	Lateral control	97
A.2.3	Optimization Method	97
A.2.4	Optimal set of gains	99
	APPENDIX B – UGV MODELING	101
B.1	Cruise	101
B.2	Steering	103
B.3	Simulink Blocks Description	104
	C – AUDI A6 SPECIFICATIONS	109
	D – PENGUIN BE UAV SPECIFICATIONS	110

List of Figures

Figure 1 – The landing maneuver.	24
Figure 2 – Penguin BE UAV.	25
Figure 3 – The landing platform.	26
Figure 4 – UAV nonlinear model.	28
Figure 5 – UGV nonlinear model.	29
Figure 6 – Roll ϕ , Pitch θ , and Yaw ψ angles, as defined in relation to a body-fixed coordinate frame.	30
Figure 7 – Definition of forces, moments and velocity components in a body-fixed coordinate frame.	30
Figure 8 – Penguin BE dynamic modes.	34
Figure 9 – Car poles	35
Figure 10 – Current longitudinal control architecture.	39
Figure 11 – Proposed longitudinal control architecture.	40
Figure 12 – Current lateral control architecture.	40
Figure 13 – Proposed lateral control architecture with runway centerline correction terms.	41
Figure 14 – Proposed lateral control architecture without runway centerline correction terms.	41
Figure 15 – Flare Law illustration.	44
Figure 16 – Lateral view of the geometrical conditions of the state machine.	45
Figure 17 – Rear view of the geometrical conditions of the state machine.	45
Figure 18 – State Transition Diagram.	46
Figure 19 – Basic Feedback Loop.	47
Figure 20 – Discrete Wind Gust Model.	49
Figure 21 – Loop shape of the longitudinal dynamics of the current architecture before optimization.	52
Figure 22 – Loop shape of the longitudinal dynamics of the proposed architecture before optimization.	53
Figure 23 – Loop shape of the lateral dynamics of the current architecture before optimization.	53
Figure 24 – Loop shape of the lateral dynamics of the proposed architecture before optimization.	54
Figure 25 – Actions of a PID controller.	55
Figure 26 – Cost function versus number of iterations for current architecture.	57
Figure 27 – Cost function versus number of iterations for proposed architecture.	58
Figure 28 – Pole-Zero plot for the current architecture.	59
Figure 29 – Pole-Zero plot for the proposed architecture	60

Figure 30 – Time-Domain Requirements.	62
Figure 31 – Landing maneuver without wind gust.	63
Figure 32 – Landing maneuver with wind gust.	64
Figure 33 – 3D representation of a landing maneuver.	66
Figure 34 – Distributions of the random inputs.	68
Figure 35 – A typical signal output from a Monte Carlo experiment.	69
Figure 36 – A typical empirical Probability Density Function.	71
Figure 37 – Feedback loop.	71
Figure 38 – Gain and sensitivity functions curves for a typical loop transfer function. . .	72
Figure 39 – Illustration of the Waterbed Effect.	74
Figure 40 – Nyquist curve of loop transfer function showing the graphical interpretation of maximum sensitivity.	74
Figure 41 – Loop shape longitudinal current architecture after optimization.	75
Figure 42 – Loop shape longitudinal proposed architecture after optimization.	76
Figure 43 – Loop shape lateral current architecture after optimization.	77
Figure 44 – Loop shape lateral proposed architecture after optimization.	77
Figure 45 – Resulting distributions from the Monte Carlo simulations of the current architecture with linear models, without time delay.	78
Figure 46 – Resulting distributions from the Monte Carlo simulations of the proposed architecture with linear models, without time delay.	79
Figure 47 – Overshoot current and proposed architectures.	79
Figure 48 – Undershoot current and proposed architectures.	80
Figure 49 – ITAE current and proposed architectures.	81
Figure 50 – Time to settle after a gust in the longitudinal direction for the current and proposed architectures.	82
Figure 51 – Time to settle after a gust in the lateral direction for the current and proposed architectures.	82
Figure 52 – Undershoot for current and proposed architectures with time delay.	83
Figure 53 – Successive loop closure design.	90
Figure 54 – Successive loop closure design with inner loop modeled as a unity gain. . .	90
Figure 55 – Bandwidth as seen from a Bode plot.	90
Figure 56 – Lateral Autopilot.	91
Figure 57 – Force equilibrium in a steady turn.	92
Figure 58 – Pitch angle inner loop.	93
Figure 59 – Thrust inner loop.	93
Figure 60 – Longitudinal Autopilot.	94
Figure 61 – Car cruise control.	95
Figure 62 – The s-plane.	96
Figure 63 – UGV linear response.	100

Figure 64 – UAV linear response.	100
Figure 65 – Torque curve.	102
Figure 66 – Bicycle Model.	104
Figure 67 – UGV Simulink mask.	105
Figure 68 – UGV engine equation.	106
Figure 69 – UGV air resistance equation.	107
Figure 70 – UGV rolling resistance equation.	107
Figure 71 – UGV lateral dynamics.	108

List of Tables

Table 1 – Eigenvalues of the Penguin BE UAV.	34
Table 2 – Values tuned with the optimization method for the current architecture. . . .	56
Table 3 – Values tuned with the optimization method for the proposed architecture. . .	57
Table 4 – Landing maneuver with the current architecture without disturbance.	63
Table 5 – Landing maneuver with the proposed architecture without disturbance. . . .	64
Table 6 – Landing maneuver with the current architecture in the presence of disturbances.	65
Table 7 – Landing maneuver with the proposed architecture in the presence of disturbances.	65
Table 8 – Monte Carlo simulation results for overshoot in the longitudinal direction. . .	70
Table 9 – Stability margins before and after optimization for the longitudinal dynamics.	76
Table 10 – Time-domain Specifications.	95
Table 11 – Second order relations	96
Table 12 – PID parameters action	96
Table 13 – Values tuned with the optimization method for the UAV linear model.	99
Table 14 – Values tuned with the optimization method for the UGV linear model.	99
Table 15 – Audi A6 Specifications.	109
Table 16 – Structural Specifications	110
Table 17 – Performance Specifications	110

List of abbreviations and acronyms

DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center)
HALE	High Altitude, Long Endurance
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
TECS	Total Energy Control System
IMU	Inertial Measurement Unit
LAN	Local-Area Network
6DOF	Six-Degree-of-Freedom
MPC	Model Predictive Control

List of symbols

u_a, v_a, w_a	Linear velocities along the x, y, and z axes for the UAV, respectively, in the body frame
u_g, v_g	Linear velocities along the x and y axes, respectively, for the UGV, in the body frame
ϕ_a, θ_a, ψ_a	Roll, pitch and yaw angles of the UAV, respectively
p_a, q_a, r_a	Roll, pitch and yaw rates of the UAV, respectively
ψ_g	Yaw angle of the car model
ω	Engine angular velocity
h	Aircraft altitude
γ	Flight path angle
χ	Course angle
V_a	Airspeed
V_k	Groundspeed
δ_a	Aileron deflection
δ_r	Rudder deflection
δ_e	Elevator deflection
δ_τ	Throttle setting
ψ_g	UGV yaw angle
λ_g	Angle of the velocity vector of the center of mass with respect to the longitudinal axis of the UGV
x_{UAV}	UAV state vector
u_{UGV}	UAV input vector
x_{UGV}	UGV state vector
u_{UGV}	UGV input vector

1 Introduction

This work is based on the author's activities at the Institute of Robotics and Mechatronics in the German Aerospace Center (DLR). The effort is included in a broader scope: Development of new technologies to improve safety and operational availability of High Altitude Long Endurance (HALE) unmanned platforms. The specific problem addressed in this thesis was approached by the DLR's Flying Robots Research Group in the scope of cooperative landing, where an Unmanned Aerial Vehicle (UAV) is intended to land on top of an Unmanned Ground Vehicle (UGV). This chapter explains the context of the work, defines its goals, and clarifies the main contribution while summarizing the content of the subsequent chapters.

1.1 Application context

In recent times, UAVs have been having a strong presence in civil and academic research applications. With the idea of improving its availability and range of applications, some challenges arise. They are mostly related to dealing with characteristics such as extremely lightweight design, battery power, large wingspan, fragile structures, and sensitivity to environmental disturbances.

Remarkable examples are HALE platforms, which are a special kind of light-weight, fixed-wing aircraft designed to operate in the stratosphere. It benefits from a calm atmosphere with the possibility of solar power generation. They have great potential to perform some tasks that are currently covered by satellites, including aerial photography, communication networks, surveillance, and atmospheric research. Common advantages include sparing the trouble and expenses of a rocket launch, orbit independence, and deployment to different locations whenever necessary. Maintenance costs (e.g. repair and replacements) are obviously greatly reduced when compared to a spacecraft devoted to the same applications.

Researchers at DLR proposed the approach of removing aircraft's landing gear as a means of reducing its weight. However, this imposed the need of using a moving platform as an asset to perform landing of an UAV without landing gear on top of an UGV. Under this concept, successful cooperative landings have been achieved with the demonstrator setup of Figure 3. The setup consists of a small UAV, the Penguin BE, and a semi-autonomous car.

1.2 Goal and objectives

The goal of this work is to compare two cooperative control architectures applied to the landing of an UAV on top of an UGV. Intermediate objectives include:

- Familiarization with vehicle models.
- Control design of individual vehicles.
- Cooperative control design for each architecture.
- Definition and application of comparison criteria.

This work was completely executed in a virtual simulation environment (MATLAB/Simulink). It allowed us to reproduce time delays due to wireless communication and wind disturbances in the simulation models.

1.3 A brief project review

From the project start, some years ago, to the present time, some conference articles and theses have been developed and published concerning the DLR's initiative of landing HALE platforms.

In Laiacker et al. (2013), vision aided landing was approached for detection and tracking of runways. A complete 6 DOF nonlinear model of the Penguin BE UAV as well as the flight controller design was developed by Balmer (2015) including the Total Energy Control System (TECS). The cooperative architecture was first introduced by Muskardin et al. (2016) together with the first simulation results. Subsequently, Muskardin et al. (2016) presented additional developments and the results of the first landing experiments. A careful linear analysis of the cooperative system and the cooperative control problem along with the investigation of the results of various landing experiments was performed by Persson (2016). Ultimately, Muskardin et al. (2017) added new improvements on the application.

1.3.1 The cooperative landing maneuver

The cooperative landing maneuver is divided into six phases, as shown in Figure 1:

Initial approach

The aircraft is at an altitude of approximately 100m above ground level flying aerodrome circuits, the maneuver starts when the operator gives the command for landing. The initial approach brings the UAV closer to the stationary UGV.

High precision approach

Once a certain distance is reached, the UAV flies to the landing start point in high precision mode (2) in order to reach it with a desired state (position and velocity).

Guided descent

Only once the landing start point is reached successfully (requirements on UAV state are met) the UGV starts accelerating to match the UAV's velocity and the cooperative part of the maneuver starts with the guided descent phase (3).

Final descent and touchdown

At this moment, both vehicles manage to align themselves with sufficient precision and the final descent takes place. When the aircraft hits the net (used to damp any undesired collision), the touchdown is accomplished.

Deceleration to full stop

In the last moments of the landing, the car reduces its velocity until a full stop.

Ground lock

Assures that the aircraft is secured on the net with a mechanical locking device.

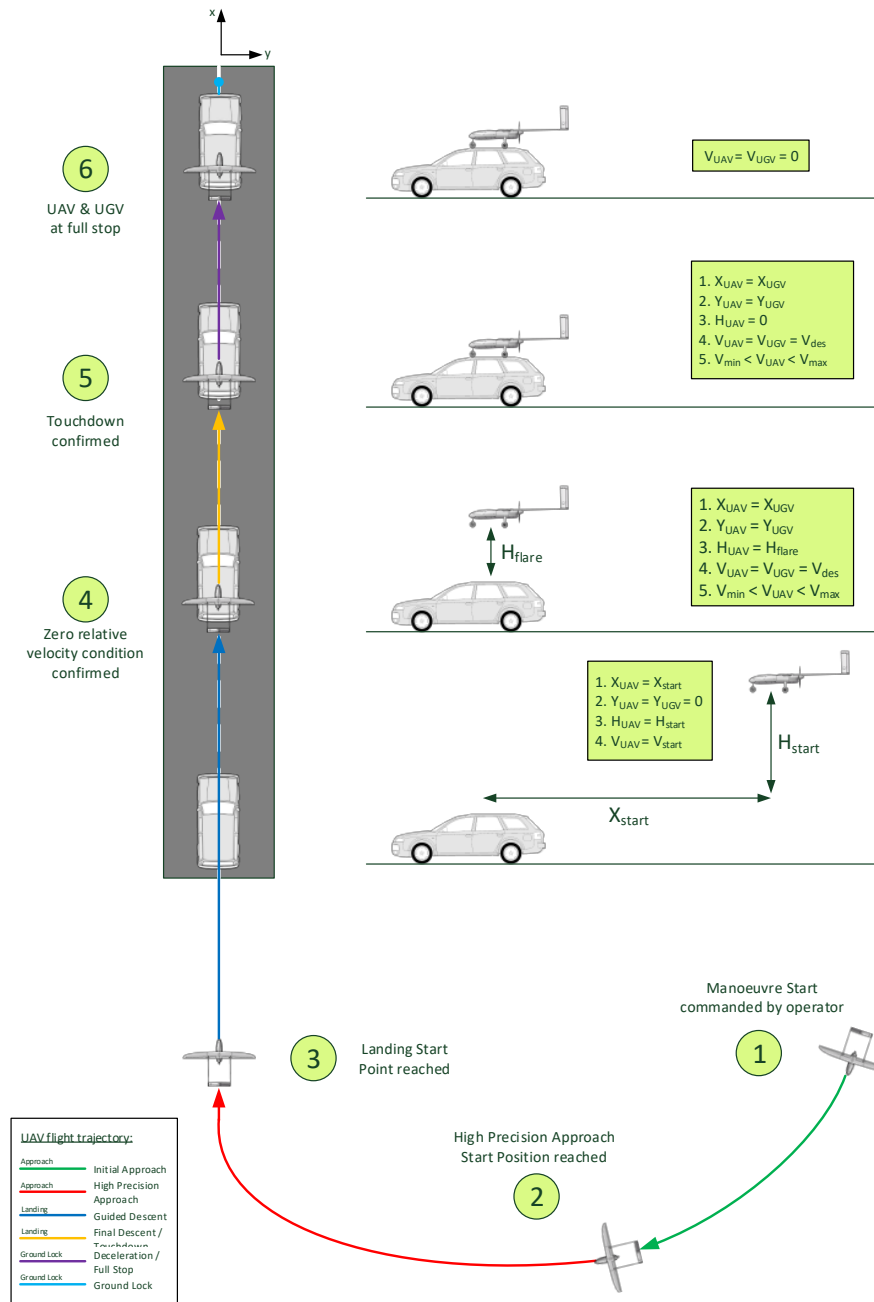


Figure 1 – The landing maneuver.

Source: (MUSKARDIN et al., 2017).

1.3.2 Cooperative Control

In a cooperative control problem, two or more dynamic systems exchange information while connected in a network aiming at the completion of a common goal. For the cooperative landing both vehicles correct for differences in position, velocity, and course in the longitudinal and lateral directions. This alignment is necessary to reach a safe landing condition.

Cooperative control of dynamic systems has emerged from the developments in commu-

nication technology and it is still a relatively new research field with a wide range of applications. Some examples of the use of cooperative control to solve real world problems include the landing of the SpaceX Falcon 9 Rocket on the dronship, aerial refueling, formation flight, and spacecraft docking. For more information regarding general applications of cooperative control, please refer to Murray (2007).

1.3.3 UAV platform overview: Penguin BE

The UAV used in this application is a commercial product¹ that has been instrumented and adapted accordingly for this application. It has a wingspan of $3.3m$ and a maximum takeoff mass of $21.5Kg$. The onboard instrumentation includes the Novatel² Flex-Pak6 RTK-GPS for accurate measurements of latitude, longitude, and altitude at a sampling rate of $20Hz$. The LORD MicroStrain³ 3DM-Gx3-25 inertial measurement unit (IMU) measures linear accelerations and angular velocities/orientation at a sampling rate of $250Hz$. The Swiss Air-Data⁴ PSS-8 pitot tube, provides airspeed and temperature information, at a $20Hz$ sampling rate and an optical camera (Allied Vision Prosilica GC1380). The aircraft communicates with the ground station through a wireless local-area network (WLAN). For additional details, please refer to Appendix D.



Figure 2 – Penguin BE UAV.

Source: UAV Factory.

1.3.4 UGV landing platform overview: a modified passengers car

The ground vehicle is a modified passengers car equipped with the same RTK-GPS and IMU as the UAV, a Graphical User Interface (GUI), and a landing platform consisting of an aluminum frame and a net attached to the car roof. The platform is $4m$ long and $5m$ wide, which gives Penguin a $1m$ offset space in each direction while located in the middle.

¹ <<http://www.uavfactory.com/>>

² <<http://www.novatel.com/>>

³ <<http://www.microstrain.com/>>

⁴ <<http://www.swiss-airdata.com/>>



Figure 3 – The landing platform.

Source: (MUSKARDIN et al., 2017).

1.4 Outline of thesis

Chapter 2 - Vehicle Models

Introduces the vehicle models.

Chapter 3 - Cooperative Control Architectures

Describes the cooperative control architecture as implemented during the last successful landing experiments and presents the proposed modifications.

Chapter 4 - Cooperative Control Design

Explains the details which are relevant to the cooperative controller design, like procedure and the typical performance metrics. Additionally, it compares the performance of both architectures.

Chapter 5 - Sensitivity Analysis

Describes the concept of the employed sensitivity analysis in time and frequency domains.

Chapter 6 - Results

Presents robustness analysis results.

Chapter 7 - Conclusion and Outlook

Summarizes the importance of the achieved results and suggests activities for future works.

2 Vehicle Models Review

The cooperative control system couples the vehicles considering their individual closed-loop dynamics. It is not in our scope to design controllers for the vehicles individually. But it is highly desirable to understand their dynamic models. In the case of the Penguin BE, a complete 6 DOF simulation model was created by Balmer (2015) and is shown here. The aircraft dynamics equations, however, are omitted for the sake of brevity. In the case of the ground vehicle, we describe the dynamic principles used in the corresponding UGV simulation model.

2.1 UAV Model

Figure 4 shows the Simulink® implementation of the 6 DOF model for the Penguin BE UAV. It contains aerodynamics, equations of motion, atmospheric models, propulsion models, support to wind disturbances, and turbulence models. The building blocks of the aircraft model are inherited from the AeroSim Blockset library as described in Balmer (2015).

The dynamic principles of the referred building blocks are well documented in (UNMANNED DYNAMICS), which is highly recommended to the interested reader. Figure 4 may be interpreted as a reference for the inherent model relationships. The internal signal flow is not relevant to our application, since it was sufficiently verified in previous works. Thus, this model is seen as a plant which interfaces with an external control system by means of control/disturbances inputs and states/sensors outputs.

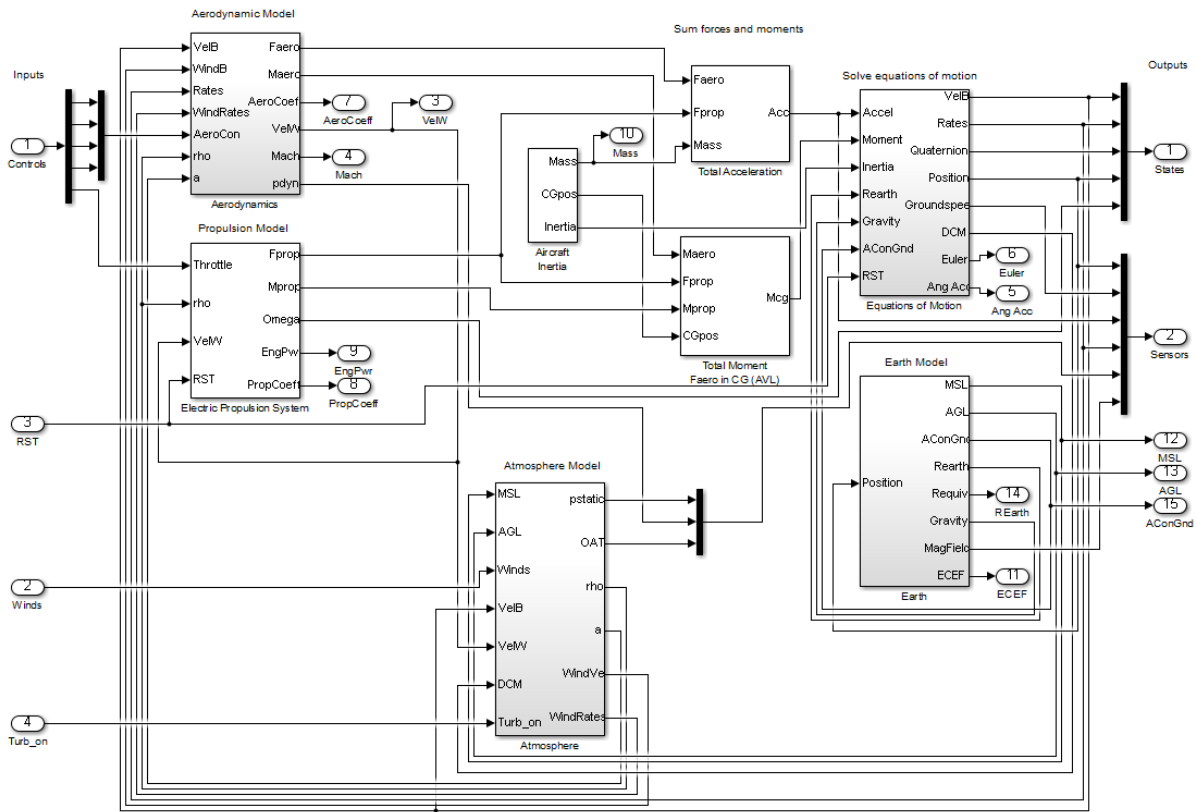


Figure 4 – UAV nonlinear model.

Source: (BALMER, 2015).

2.2 UGV Model

The nonlinear car model was implemented in Simulink® based on a dynamic model for the longitudinal motion (ASTRÖM; MURRAY, 2010) and a kinematic model, within a simplification called bicycle model for the lateral motion (KONG et al., 2015). The parameter values were taken from the datasheet of the manufacturer. The full nonlinear model can be seen in Figure 5, with the different blocks representing the engine, aerodynamic drag and rolling friction.

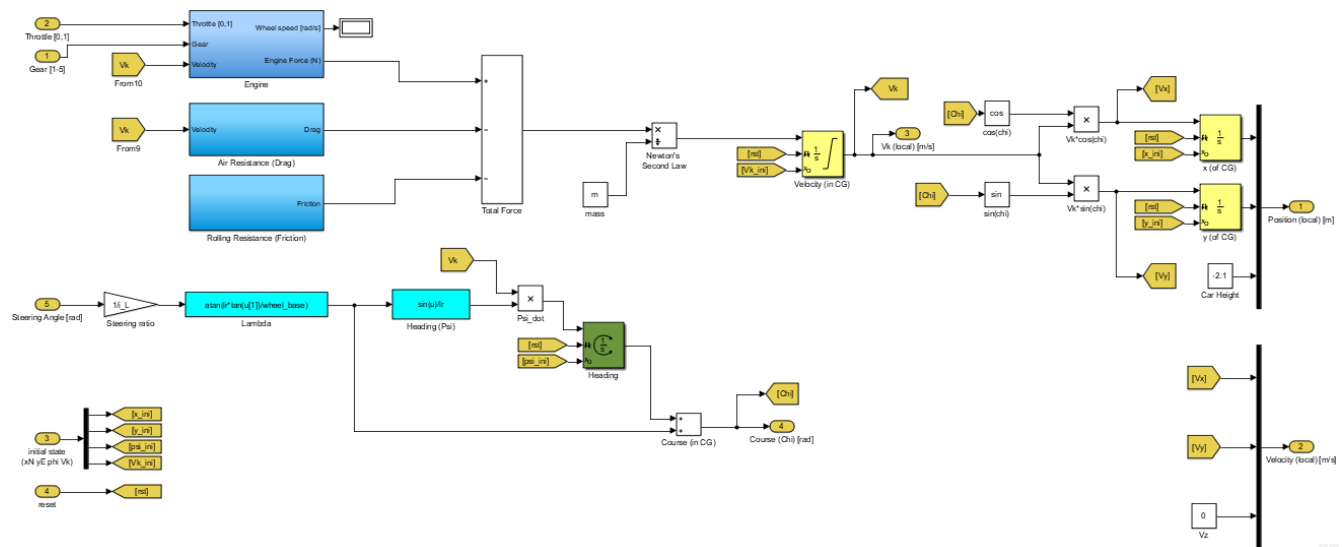


Figure 5 – UGV nonlinear model.

2.3 Linearization

For analysis and control design it is common practice to deal with linear models as there are a great variety of techniques available to deal with such models. Since our models are nonlinear, we obtained their linear versions by means of numerical (UAV) and analytical (UGV) linearization. The interested reader may refer to Close, Newell and Frederick (2002) in order to review the basic theory on linearization of nonlinear systems for control design purposes. A resulting linearized model is limited to describe the local behavior of the original nonlinear system. The subsequent subsections describe the linear models obtained from our nonlinear models for both vehicles.

2.3.1 Linear aircraft dynamics

The aircraft dynamics is often decoupled in longitudinal and lateral dynamics for linear analysis. These representations cover the following dynamic modes: short period, phugoid, spiral, dutch roll, and roll subsidence. This section describes the variables that form the aircraft state-space model split into longitudinal and lateral dynamics. Additional information about the procedure for deriving a linear model for small aircraft, including a detailed description of all the equations of motion and corresponding aerodynamic coefficients can be found in Beard and McLain (2012) and Cook (2012).

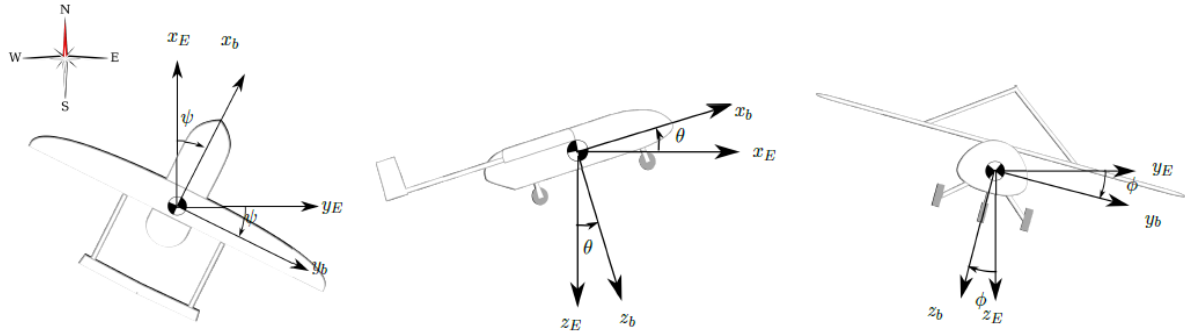


Figure 6 – Roll ϕ , Pitch θ , and Yaw ψ angles, as defined in relation to a body-fixed coordinate frame.

Source: (PERSSON, 2016).

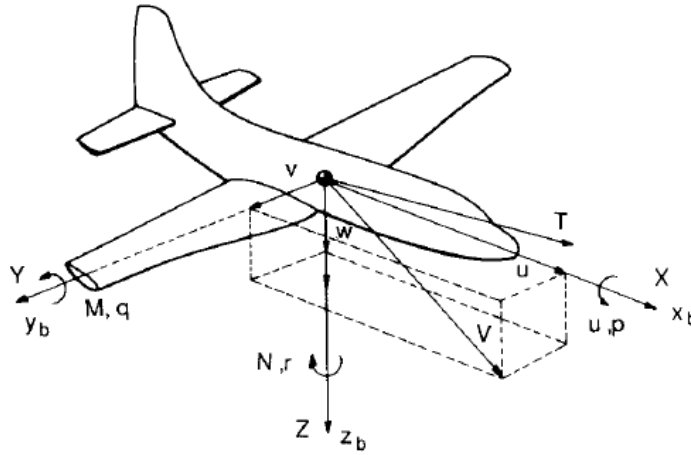


Figure 7 – Definition of forces, moments and velocity components in a body-fixed coordinate frame.

Source: (NELSON et al., 1998).

2.3.1.1 Longitudinal model

For the longitudinal dynamics, the aircraft states are given by

$$x_{lon} = (u_a, w_a, q_a, \theta_a, \omega_a)^T$$

where u_a is the velocity component along the roll axis, w_a is the velocity component along the yaw axis, q_a is the pitch rate (angular rate about pitch axis), θ_a is the pitch angle, and ω_a is the motor angular velocity.

The input vector is defined as

$$u_{lon} = (\delta_e, \delta_\tau)^T$$

where δ_e is the elevator deflection angle and δ_τ is the throttle setting. The numerical linearization performed by Persson (2016) for the Penguin BE UAV leads to the following state-space matrices

$$\begin{bmatrix} \dot{u}_a \\ \dot{w}_a \\ \dot{q}_a \\ \dot{\theta}_a \\ \dot{\omega}_a \end{bmatrix} = \begin{bmatrix} -0.10 & 0.39 & -1.4 & -9.8 & 0.006 \\ -0.64 & -3.6 & 22 & -0.6 & 0 \\ 0.19 & -2.8 & -5.6 & 0 & -0.001 \\ 0 & 0 & 1 & 0 & 0 \\ 21 & 1.2 & 0 & 0 & -2.6 \end{bmatrix} \begin{bmatrix} u_a \\ w_a \\ q_a \\ \theta_a \\ \omega_a \end{bmatrix} + \begin{bmatrix} 0.38 & 0 \\ -7.3 & 0 \\ -65 & 0 \\ 0 & 0 \\ 0 & 2027 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_\tau \end{bmatrix} \quad (2.1)$$

2.3.1.2 Lateral model

For the lateral state-space equations, the aircraft states are given by

$$x_{lat} = (v_a, p_a, r_a, \phi_a, \psi_a)^T$$

where v_a is the velocity component along the pitch axis, p_a is the roll rate (angular rate about roll axis), r_a is the yaw rate, ϕ_a is the roll angle, and ψ_a is the yaw angle.

The input vector is defined as

$$u_{lat} = (\delta_a, \delta_r)^T$$

where δ_a is the aileron deflection and δ_r is the rudder deflection. Analogous to the longitudinal dynamics, the corresponding numerically linearized lateral dynamics can be summarized as follows:

$$\begin{bmatrix} \dot{v}_a \\ \dot{p}_a \\ \dot{r}_a \\ \dot{\phi}_a \\ \dot{\psi}_a \end{bmatrix} = \begin{bmatrix} -0.28 & 1.9 & -21 & 9.8 & 0 \\ -0.40 & -14 & 2.5 & 0 & 0 \\ 1.3 & -2.1 & -1.2 & 0 & -0.001 \\ 0 & 1 & 0.08 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_a \\ p_a \\ r_a \\ \phi_a \\ \psi_a \end{bmatrix} + \begin{bmatrix} 0.69 & 4.5 \\ -131 & -3.9 \\ -19 & -23 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \quad (2.2)$$

2.3.2 Ground Vehicle

The linear state-space model for the car was derived analytically, solving the partial derivatives of the Jacobian matrices.

2.3.2.1 Cruise

As explained in Appendix B, the differential equations for the longitudinal motion of the car are:

$$m \frac{dv}{dt} = F - F_d \quad (2.3)$$

$$f_1 = \frac{dv}{dt} = \frac{\alpha_n T(\alpha_n v) \delta_\tau - \frac{1}{2} \rho C_d A v |v| - mg \sin(\theta) - mg C_r \text{sgn}(v)}{m} \quad (2.4)$$

where F is the driving force, F_d the sum of the forces opposing the movement, v^1 is the velocity in the forward direction, θ is the slope of the road, T is the torque about the wheel axis, δ_τ is the throttle setting, m is the mass of the car, g is the acceleration of gravity, α_n is the gear ratio, C_r is the rolling coefficient, and C_d is the drag coefficient.

For a complete description of all variables and steps taken in the mathematical modeling procedure for the car, please consult Appendix B.

- For the car longitudinal dynamics, any feasible speed corresponds to an equilibrium point.
- Choose a reference speed $v_{ref} > 0$, and solve for $\frac{dv}{dt} = 0$ with respect to δ_τ , assuming a horizontal road ($\theta = 0$).²

$$0 = \alpha_n T(\alpha_n \bar{v}) \bar{\delta}_\tau - \frac{1}{2} \rho C_d A \bar{v}^2 - mg C_r$$

$$\bar{\delta}_\tau = \frac{\frac{1}{2} \rho C_d A \bar{v}^2 + mg C_r}{\alpha_n T(\alpha_n \bar{v})}$$

$$f_2 = \frac{dx}{dt} = v \cos(\psi + \lambda) \quad (2.5)$$

2.3.2.2 Steering

As explained in Appendix B, the differential equations for the lateral dynamics are:

$$f_3 = \frac{dy}{dt} = v \sin(\psi + \lambda) \quad (2.6)$$

$$f_4 = \frac{d\psi}{dt} = \frac{v}{l_r} \sin(\lambda) = \frac{v}{l_r + l_f} \tan(\delta) \quad (2.7)$$

¹ $v = u_g$ = velocity component in the longitudinal direction.

² $\bar{v}, \bar{\delta}_\tau$ = velocity and throttle at equilibrium.

$$f_5 = \lambda(\delta) = \arctan\left(\frac{l_r}{l_r + l_f} \tan(\delta)\right) \quad (2.8)$$

where ψ is the yaw angle, λ is the angle of the velocity vector of the center of mass with respect to the longitudinal axis of the car, δ is the steering wheel deflection, l_r is the distance between the center of mass and the rear wheel, and l_f is the distance between the center of mass and the front wheel of the car.

If the equilibrium point is considered as: $x_e = [v \ x \ y \ \psi \ \chi]^T = [21 \ 100 \ 0 \ 0 \ 0]^T$

The Jacobians for the A and B matrix are calculated substituting the car parameters from Appendices B and C, into the corresponding partial derivatives evaluated at the equilibrium point. Both matrices are presented below, where f is the throttle input and δ is the steering wheel deflection.

$$\begin{bmatrix} \dot{u}_g \\ \dot{x}_g \\ \dot{y}_g \\ \dot{\psi}_g \\ \dot{\lambda}_g \end{bmatrix} = \begin{bmatrix} -0.00781 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 21 & 21 \\ 0 & 0 & 0 & 0 & 15.22 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_g \\ x_g \\ y_g \\ \psi_g \\ \lambda_g \end{bmatrix} + \begin{bmatrix} 1.7590 & 0 \\ 0 & 0 \\ 0 & 10.5 \\ 0 & 7.6087 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} f \\ \delta \end{bmatrix} \quad (2.9)$$

$$\begin{bmatrix} \dot{x}_g \\ x_g \\ y_g \\ \dot{y}_g \\ \chi_g \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 21 & 21 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} u_g \\ x_g \\ y_g \\ \psi_g \\ \lambda_g \end{bmatrix} \quad (2.10)$$

where u_g is the longitudinal component of the velocity vector, x_g and y_g are longitudinal and lateral position coordinates, respectively. ψ_g is the yaw angle, λ_g is the angle of the velocity vector of the center of mass with respect to the longitudinal axis of the car, and χ_g is the course angle.

2.4 Stability

Stability is a primary concern, since it means the system is not going to grow unbounded. The position of system poles indicates whether the system is stable or not. If all the poles are in the left half plane, the system is stable. If at least one pole is in the right half plane the system is unstable, thereby some control action needs to be applied to stabilize the system.

2.4.1 Aircraft Dynamic Modes

Figure 8 shows the plot of the location of the poles for the Penguin BE where the corresponding modes are highlighted. Its eigenvalues, the corresponding natural frequencies, and damping are presented in Table 1.

As shown in the last row of Table 1, the spiral mode has an eigenvalue of 0.106 and is thus unstable. Nevertheless, this is not a surprise considering that it is common to have an unstable spiral mode. In fact, it is usually easy to control it due to its large time constant.

Table 1 – Eigenvalues of the Penguin BE UAV.

Mode	Eigenvalue	$\omega_n(rad/s)$	$f(Hz)$	ζ
Short Period	$-4.61 \pm 7.86i$	9.11	4.6083	0.506
Phugoid	$-0.0307 \pm 0.5353i$	0.536	0.0308	0.0573
Dutch Roll	$-0.960 \pm 5.19i$	5.28	0.9615	0.182
Roll Subsidence	-13.4	13.4	13.3869	1.00
Spiral	0.106	0.106	-0.106	-1.00

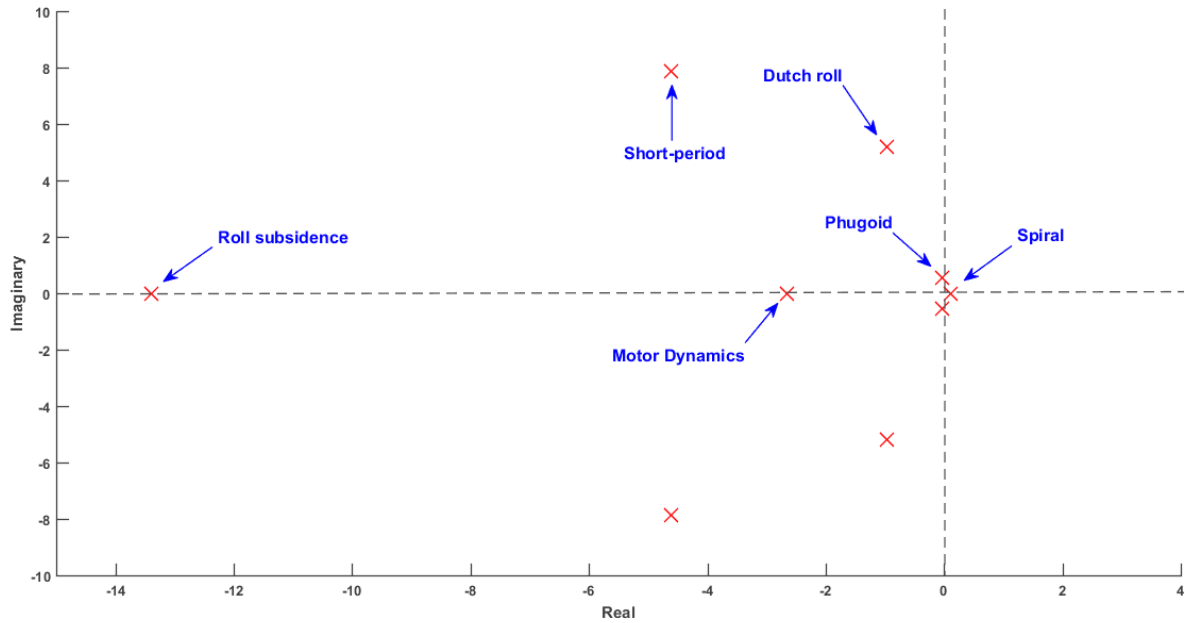


Figure 8 – Penguin BE dynamic modes.

2.4.2 Car Poles

We also rely in linear analysis to draw comments on car's stability. As shown in Figure 9 the car dynamics can be represented by four poles at the origin and a pole at -0.0078 .

Furthermore, it is by changing the poles position in the s -plane that the controller influences the system response.

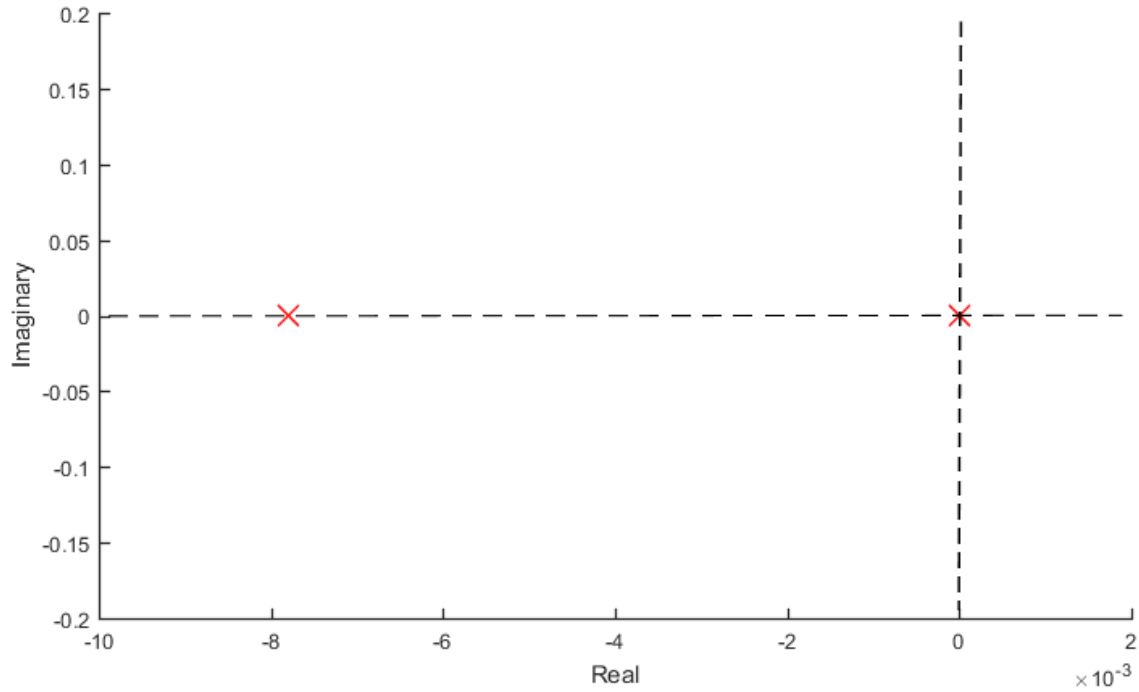


Figure 9 – Car poles.

Examples of controller synthesis for the aircraft and the ground vehicle can be found in Appendix A.

3 Cooperative Control Architectures

At this point we have the vehicles' models and respective controllers embedded in a single simulation model. It is thus time to address the actual landing problem and analyze the cooperative control architectures. Although a successful landing was already accomplished in previous experiments (MUSKARDIN et al., 2017), some issues and possibilities of improvement were identified. In this chapter, the modifications adopted in the control structure are presented along with pros and cons of each architecture.

3.1 Current Architecture

For the longitudinal motion, the current control strategy applies velocity commands as control inputs, which are proportional to position and velocity errors in the longitudinal direction (often referred as x -direction) (MUSKARDIN et al., 2016). The position measurements are taken with respect to an inertial reference frame; in which the x -axis is defined along the runway, the y -axis points to the right, and the z -axis points downwards, as illustrated in Figure 1.

In addition, course angle commands proportional to position and velocity errors in the y -direction are applied for the lateral motion. Such combination of Cartesian and Polar coordinates results in a coupling between the lateral and longitudinal motions. In this case, absolute groundspeed (scalar quantity) is commanded in the longitudinal controller, instead of just the x -component of the velocity vector.

Therefore, as lateral errors are compensated through angular (directional) motion of the vehicles, a stronger coupling of the longitudinal and lateral controllers arise for large course angles. For instance, if both vehicles are already aligned in the x -direction but not in the y -direction, a rotation of the velocity vector around the z -axis will lead to a deviation in x -position after one integration step. The longitudinal controller will try to compensate for this error by increasing the length of the velocity vector which in turn will affect the lateral control. At small course angles this coupling is negligible. Also, such coupling only applies to deviations in lateral direction. If the vehicles are aligned in y but need to correct for a deviation in x , a coupling does not occur.

3.2 Proposed Architecture

When designing the new architecture, the characteristics mentioned in the last section were taken into consideration. A new scheme has been suggested, in which the longitudinal and lateral motions are decoupled and the controller acts on the same type of variable. For

example: position error in longitudinal direction will lead to a velocity command in x -direction (and not in some direction which depends on the current lateral control action). Furthermore, an advantage of the proposed architecture is that the longitudinal and lateral control loops were made symmetric, as will be further explained in Section 3.5.2.2. This improves the intuition during the design phase.

3.3 SISO and MIMO Systems

In the case of LTI (Linear Time-Invariant) systems, a Single-Input-Single-Output (SISO) loop is represented by a transfer function as the relation between one input and a specific output. On the other hand, in Multi-Input-Multi-Output (MIMO) systems relations between multiple inputs and outputs are considered (FRANKLIN et al., 2009).

As reviewed in previous topics of this work, the dynamics of the aircraft and the car are multivariable systems by themselves. When arranged together in a cooperative architecture, the number of inputs and outputs is the sum of their individual ones. Moreover, disturbances and noise may be considered as additional inputs.

In control system simulation-assisted analysis, it is usually advantageous to represent the dynamic equations in a signal flow manner (e.g. block diagrams) with decoupled longitudinal and lateral dynamics (BLAKELOCK, 1991).

3.4 State-Space Representation

In Section 2.3 linear models for the UAV and UGV have been described. By appending the models, the total system can be represented in the form 3.1.

$$\begin{bmatrix} \dot{x}_{UAV} \\ \dot{x}_{UGV} \end{bmatrix} = \begin{bmatrix} A_{UAV} & 0 \\ 0 & A_{UGV} \end{bmatrix} \begin{bmatrix} x_{UAV} \\ x_{UGV} \end{bmatrix} + \begin{bmatrix} B_{UAV} & 0 \\ 0 & B_{UGV} \end{bmatrix} \begin{bmatrix} u_{UAV} \\ u_{UGV} \end{bmatrix} \quad (3.1)$$

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta V_x \\ \Delta V_y \end{bmatrix} = [C] \begin{bmatrix} x_{UAV} \\ x_{UGV} \end{bmatrix}$$

$$\Delta x = x_g - x_a$$

$$\Delta y = y_g - y_a$$

$$\Delta V_x = u_g - u_a$$

$$\Delta V_y = v_g - v_a$$

where the state and control vectors are

$$x_{UAV} = [u_a \ x_a \ w_a \ q_a \ \theta_a \ \omega_a \ v_a \ y_a \ p_a \ r_a \ \phi_a \ \psi_a]^T$$

$$u_{UAV} = [V_{kdes} \ \chi_{des}]^T$$

$$x_{UGV} = [u_g \ x_g \ v_g \ y_g \ \psi_g \ \lambda_g]^T$$

$$u_{UGV} = [V_{kdes} \ \chi_{des}]^T$$

As we often refer to the control architectures using block diagrams and transfer functions, one shall consider the complete dynamics (longitudinal and lateral) of the UGV and UAV as tuples of the form: $G(s) = \langle {}^oG(s), {}^aG(s) \rangle$ and $A(s) = \langle {}^oA(s), {}^aA(s) \rangle$, where ${}^oG(s)$ and ${}^oA(s)$ refer to the longitudinal portion of the UGV and UAV dynamics while ${}^aG(s)$ and ${}^aA(s)$ are their lateral counterparts.

3.5 Position Alignment and Touchdown

The final goal in this application is to have both vehicles aligned in position, velocity, and acceleration along all three axes of a given inertial reference frame (e.g local runway coordinates). For that purpose, a controller structure must be designed for each direction. The cooperative landing problem can be divided into three parts, namely, longitudinal, lateral and vertical. The longitudinal and lateral controllers must align the two vehicles and, in the moment safe condition is reached, the aircraft starts descending towards the landing platform. When considering the final landing application, this control problem becomes quite complex and there are many potential solutions with different degrees of sophistication. Due to its simplicity and disturbance rejection characteristics (LI; ANG; CHONG, 2006), the choice was for an approach similar to Lavretsky (2002)'s - using PID controllers. Additionally, the DLR's Flying Robots Group recently implemented a MPC-based controller (not in the scope of this work) for generation of optimal trajectories during the approach and descent phases until close proximity motion is reached (PAVANI, 2018). This allows an easy shaping of desired maneuver characteristics (e.g. minimization of runway length, landing time, among others). It is intended to use both controllers in future applications, as their combination could drive the system to an enhanced performance and safety.

3.5.1 Longitudinal Control

3.5.1.1 Current Architecture

Figure 10 shows the block diagram representation of the controller structure as implemented during the first landing experiments. In order to simplify the representation, the notation $G(s)$ was used for the ground vehicle while $A(s)$ was used for the aircraft. In terms of controller gains, C_{XG} represents the ground vehicle longitudinal controller and this block contains a PID controller, that is, inside the block C_{XG} there are proportional, integral and derivative terms (K_{pXG} , K_{iXG} and $K_{p\dot{X}G}$). Actually, instead of using a derivative term directly, it has been used a term proportional to the velocity error ($K_{p\dot{X}G}$), since the velocity measures are available in this application.

As can be seen, only the UGV receives control commands proportional to error in the x -position, while the UAV follows a constant velocity. Therefore, the maneuver is coupled, since the ground vehicle knows the aircraft states, but it is not cooperative according to Definition 1.3.2. A great feature of this architecture is the feeding of the UAV longitudinal velocity (V_{xUAV}) to the UGV, as it increases the system ability to react faster to disturbances; this idea was further exploited in the proposed architecture. Any disturbance analyses are performed by keeping the references Δ_{Xdes} and Δ_{Ydes} equal to zero.

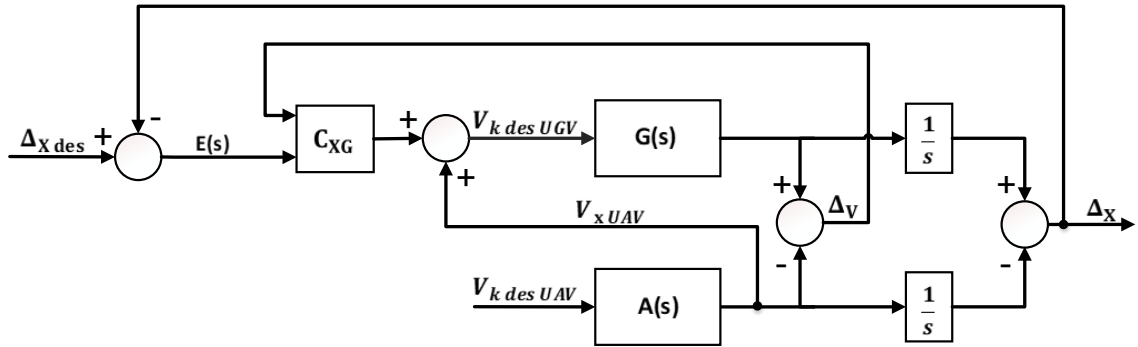


Figure 10 – Current longitudinal control architecture.

3.5.1.2 Proposed Architecture

The proposed controller structure for the longitudinal direction is illustrated in Figure 11. In this configuration, the longitudinal controller is actually decoupled from the lateral controller by commanding single axis control commands (x -position error generates x -velocity commands). The controller input is the x -component of the velocity vector instead of the V_k (groundspeed). Additionally, in this scheme both vehicles correct for differences in position. Therefore, it is cooperative.

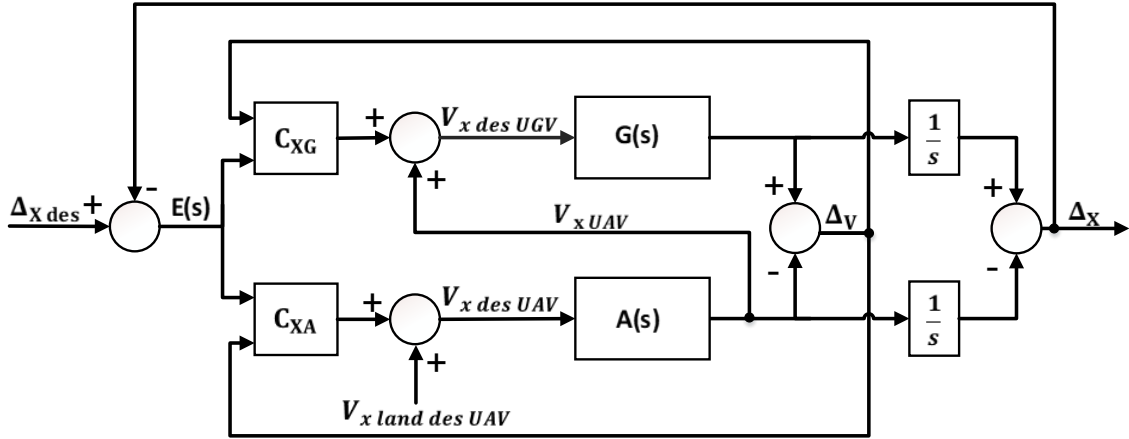


Figure 11 – Proposed longitudinal control architecture.

3.5.2 Lateral Control

3.5.2.1 Current Architecture

In the current lateral control architecture, the maneuver is also coupled, but not cooperative. The UGV remains in the center of the runway during the entire landing maneuver. The UAV is the only agent responsible for compensating lateral position errors. This scheme is represented in Figure 12, where the control input is a desired course angle χ_{des} .

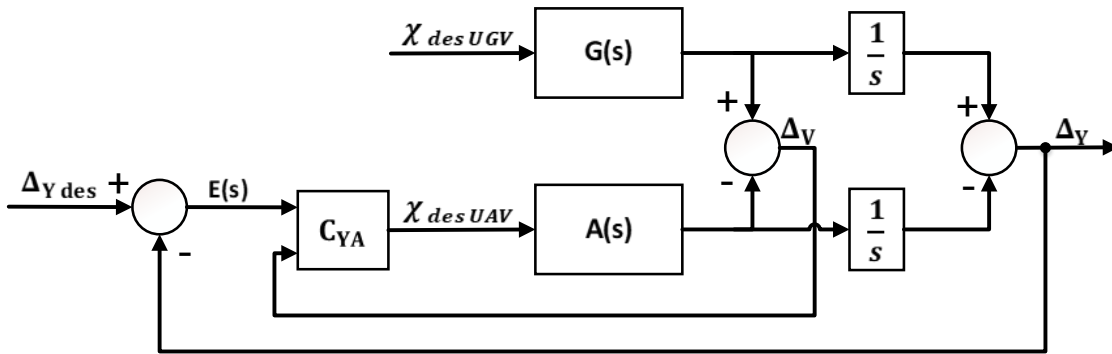


Figure 12 – Current lateral control architecture.

3.5.2.2 Proposed Architecture

In the proposed lateral control architecture both vehicles are responsible for correcting for errors in the y -direction. The control inputs for the vehicles are the y -component of the velocity vector V_{ydes} instead of the χ_{des} . As can be seen in Figures 11 and 13, in the proposed architecture, the longitudinal and lateral dynamics are symmetric. Apart from the runway center correction terms, C_{YG2} and C_{YA2} , which are only necessary in the lateral motion when it is

required the landing to happen in the center of the runway. The physical interpretation of this term is that of a spring pulling the vehicles to the runway centerline. However, since the runway where the landing experiments are to be performed is quite large (40m wide), this is not mandatory. Hence, the proposed lateral control architecture can be represented as in Figure 14.

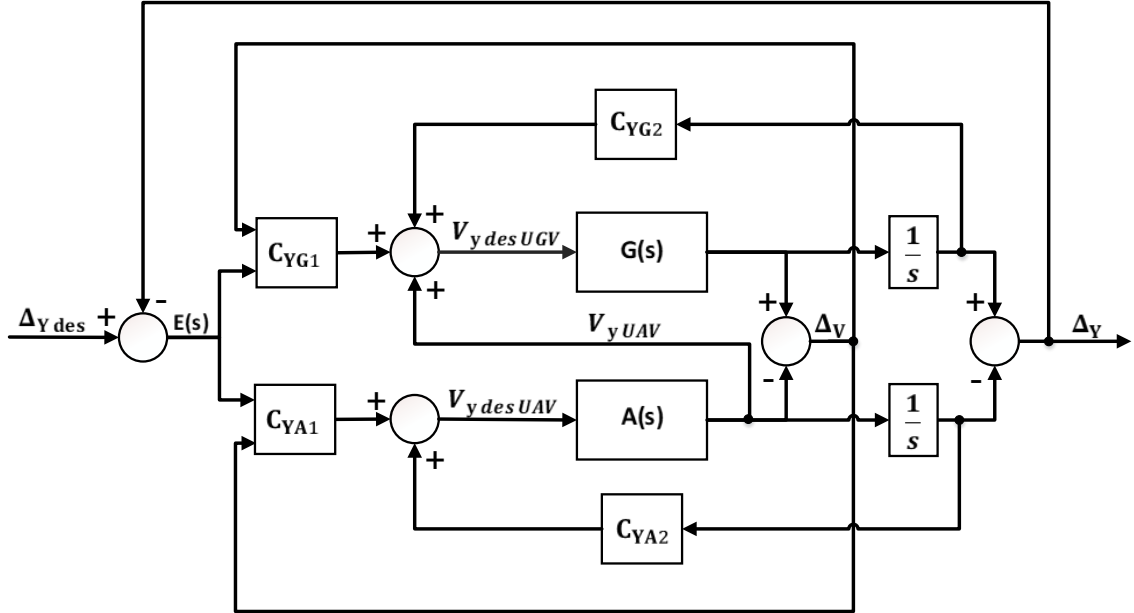


Figure 13 – Proposed lateral control architecture with runway centerline correction terms.

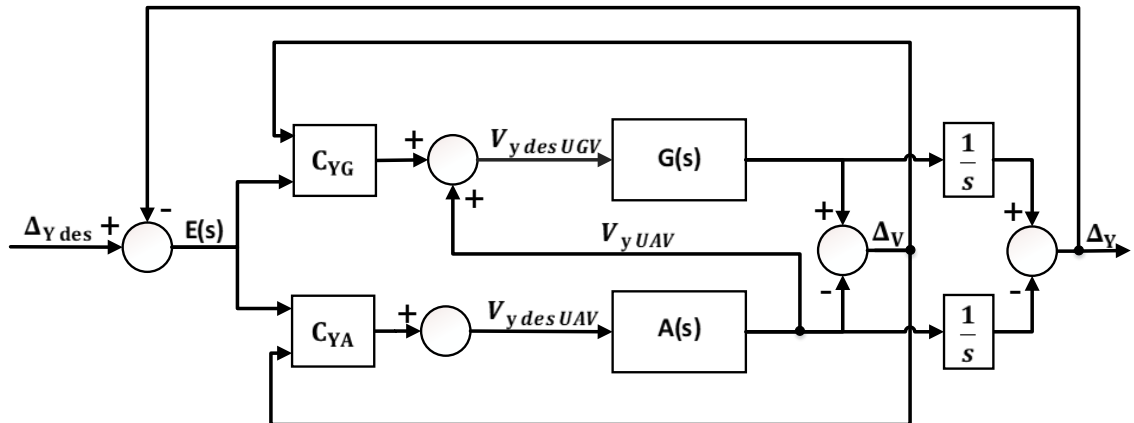


Figure 14 – Proposed lateral control architecture without runway centerline correction terms.

As briefly introduced in Section 3.2, in the proposed architecture the resulting control commands for both vehicles in the horizontal plane are calculated as

$$V_k(t) = \sqrt{V_x^2(t) + V_y^2(t)} \quad (3.2)$$

$$\tan \chi(t) = \frac{V_y}{V_x} \rightarrow \chi(t) = \arctan\left(\frac{V_y}{V_x}\right) \quad (3.3)$$

3.5.3 Vertical Control

In the last two subsections, the strategies for the longitudinal (3.5.1) and lateral (3.5.2) controllers were stated. With the vehicles aligned in the x and y axis, the next step concerns the vertical control, or descent, of the UAV. The performance of the descent depends on the performance of the lateral and longitudinal controllers because the landing cannot happen if the requirements 3.5.3.1 are not satisfied. The vertical control was not designed in this work, but it is briefly described as a means of providing a full understanding of the landing maneuver. As in this case a proposed architecture is not applicable, we recall that Persson (2016) suggested better ways to adapt the sink rate to the error in position of the vehicles. This could potentially improve the overall response, since it would prevent the system from entering undesirable modes in the state machine, as briefly described in Section 3.6. In that way, the aircraft would adjust its descent rate according to the error in position, so it would only go for the final descent and touchdown when a safe condition has been reached.

3.5.3.1 Landing Requirements

In order to achieve safe landing, several requirements must be satisfied. First of all, there are physical limitations on the system. For instance, the aircraft can not assume arbitrary sink rates \dot{h} and airspeeds V_a . Besides that, the ground vehicle can not leave the runway limits in x and y , for example, considering a runway $40m$ wide and the origin of the reference frame in its center, the car must always remain between the positions $-20m$ and $20m$ in the lateral direction. Another requirement related to the cooperative control task is the limitation on the maximum distance between the vehicles. The landing must take place within the area of the landing platform, as described in Section 1.3.4.

Thus, the requirements are:

$$V_{a \text{ land}}^{\min} \leq V_{a \text{ land}} \leq V_{a \text{ land}}^{\max} \quad (3.4)$$

$$\dot{h}_{\text{land}} \leq \dot{h}_{\text{land}}^{\max} \quad (3.5)$$

$$X_{UGV}^{\min} \leq X_{UGV} \leq X_{UGV}^{\max} \quad (3.6)$$

$$Y_{UGV}^{\min} \leq Y_{UGV} \leq Y_{UGV}^{\max} \quad (3.7)$$

$$|\Delta x| \leq \Delta x^{max} \quad (3.8)$$

$$|\Delta y| \leq \Delta y^{max} \quad (3.9)$$

The range between the minimum and maximum airspeeds of an aircraft is defined by its flight envelope as shown in Equation 3.4. Outside of this envelope safe flight is not guaranteed, as the airplane could enter stall (loss of lift) or overspeed (structural limit) conditions.

3.5.3.2 Flare Law

The descent procedure is usually divided into two parts. First, the aircraft is commanded to follow a constant flight path angle of $\gamma = -3^\circ$ glideslope. Then, at the moment it reaches the so-called *flare altitude*, it changes to a second approach: a final descending strategy called *flare*. Since any aircraft is subject to limitations at landing, there are a number of different strategies for adapting forward speed and sink rate in the literature. In this particular problem, there are strict requirements on the vehicle's relative distance. The descent controller needs to consider the relative state of the system, so it would then be possible to adapt the sink rate to this information. Figure 15 illustrates the vertical control strategy as implemented previously. In this strategy, the airplane first descends with a constant flight path angle γ (blue line) until it reaches the altitude H_{zero} when it then starts to receive variable flight path angle commands calculated according to the flare law of Equation 3.10. It is important to note that all states highlighted in the yellow boxes need to be satisfied for the sequential continuity of the maneuver.

The classic flare law currently in use has the form:

$$\dot{h}_{flare} = k(h + h_B) \quad (3.10)$$

where k and h_B are parameters which are chosen according to the landing objectives.

$k \rightarrow$ adjustment of flare curvature or steepness;

$h_B \rightarrow$ altitude bias for adjusting desired touchdown vertical speed.

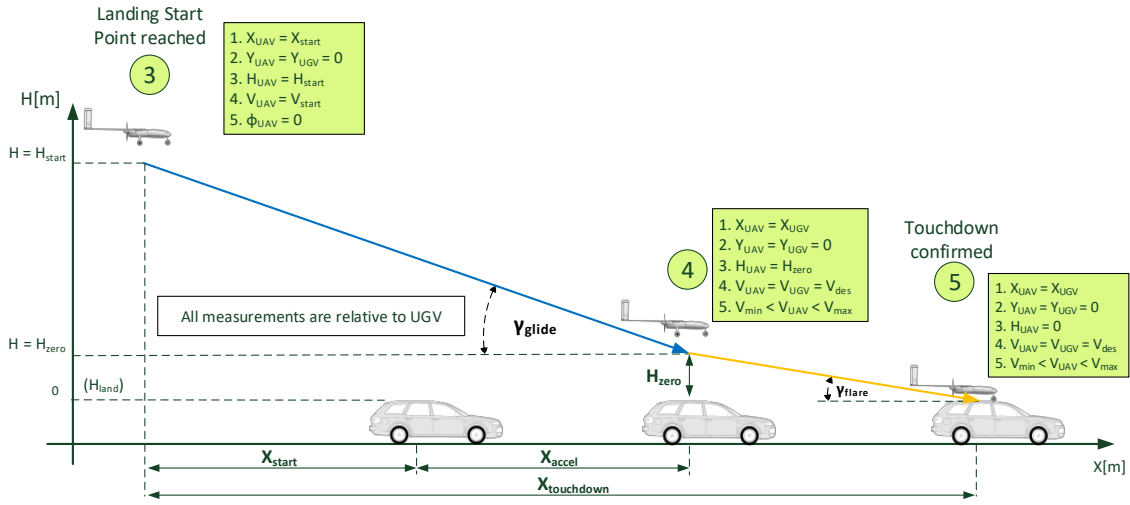


Figure 15 – Flare Law illustration.

Source: (EC-SAFEMOBIL, 2016).

3.5.3.3 Current Architecture

As previously mentioned, the vertical control approach currently in use consists of two parts (MUSKARDIN et al., 2017). First, a constant flight path angle $\lambda = -3^\circ$ is commanded and when the flare altitude of $5m$ is reached, a classical flare law in the form of Equation 3.10 is activated, which does not consider the relative state of the vehicles. It has been observed in previous landing experiments that sometimes the aircraft descends towards a landing while there is still a large error in position. Due to this behavior, the system enters undesirable modes in the state machine, as will be further explained in Section 3.6. This forces the aircraft to climb and move away from the UGV, thus unnecessarily delaying the landing maneuver. Sometimes it makes it impossible to land, since there is a limit on the length of the runway. This strategy is illustrated in Figure 15.

3.6 Finite State Machine

A finite state machine (FSM) is a method of modeling a system with a limited number of operating modes (states). The changes (transitions) between subsequent states in a FSM are event driven; while different dynamic behaviors are associated to each FSM state. These transitions promote the switch from one state to another after a specific logic condition is reached. As mentioned previously in this chapter, a state machine has been developed by Muskardin et al. (2016) to implement the complete cooperative landing focusing on increasing safety and providing high-level mission control for the application.

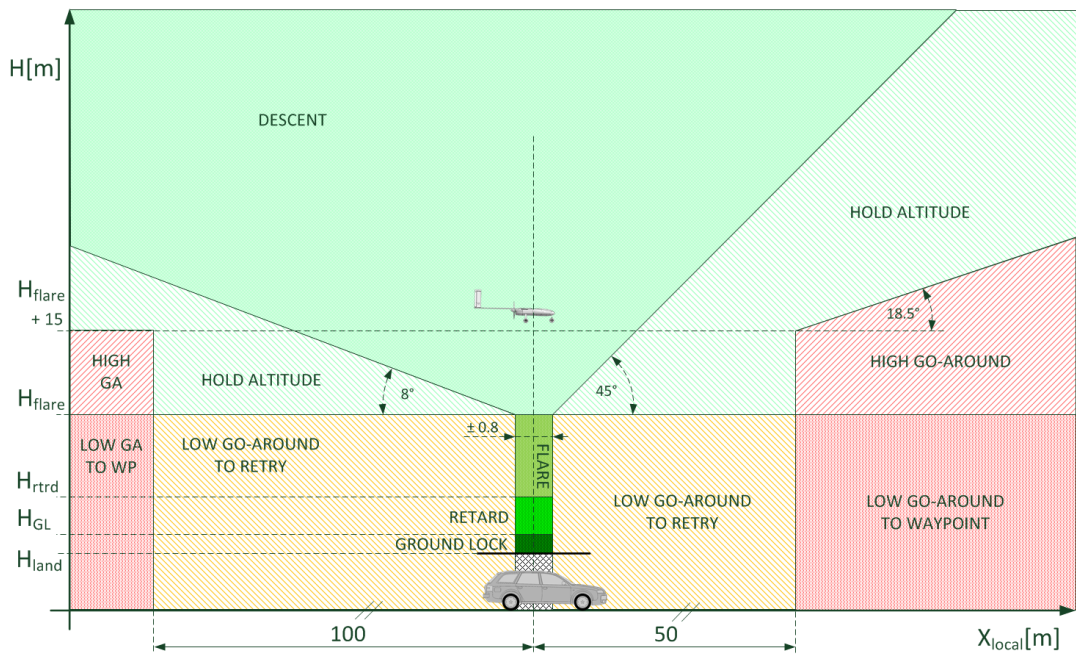


Figure 16 – Lateral view of the geometrical conditions of the state machine.

Source: (EC-SAFEMOBIL, 2016).

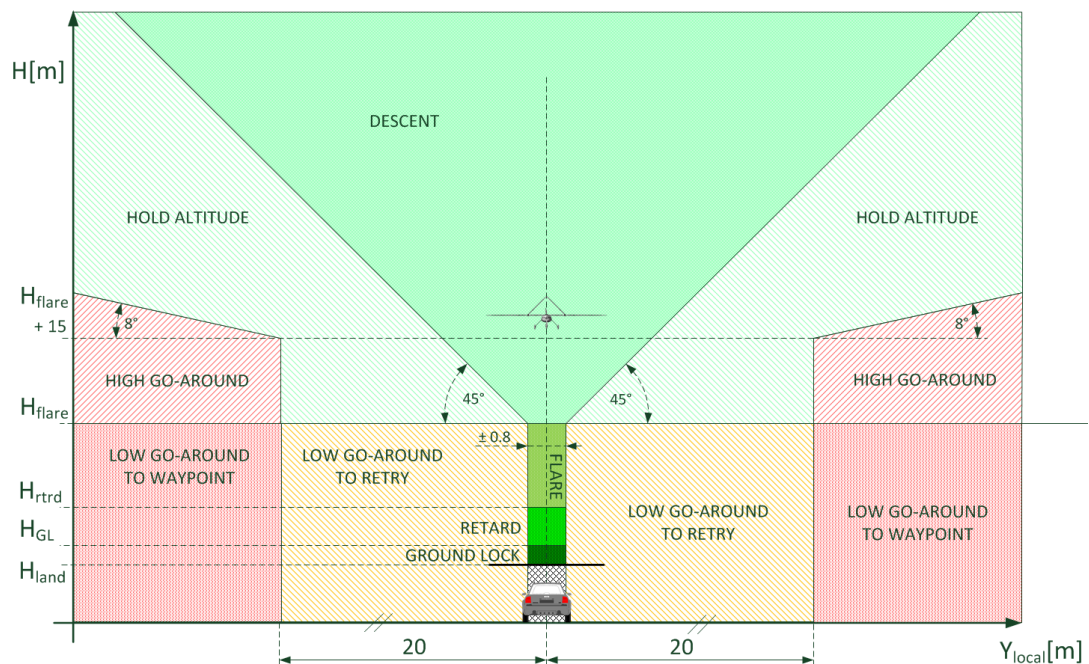


Figure 17 – Rear view of the geometrical conditions of the state machine.

Source: (EC-SAFEMOBIL, 2016).

Figures 16 and 17 illustrate the different zones and their corresponding actions. The switching conditions of the FSM are tied to the definition of these zones. Figure 18 shows the state transition diagram for the cooperative system. The states are represented by rectangles

and the arrows indicate a transition between states, the condition that enables the transition is located next to the corresponding arrow. In this case, for example, a transition to the state of flare happens when the aircraft reaches an altitude lower than H_{flare} (lower left corner).

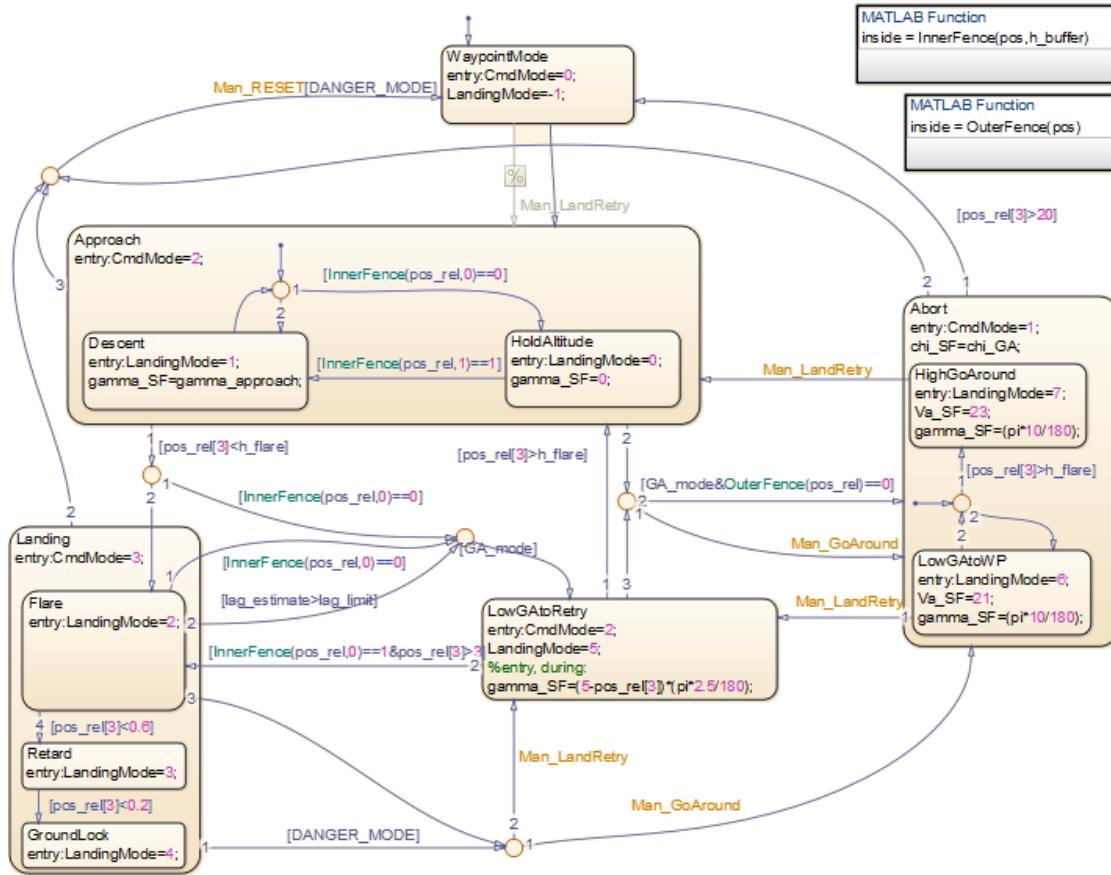


Figure 18 – State Transition Diagram.

Source: (EC-SAFEMOBIL, 2016).

If the UAV remains in the green areas (corresponding to descend, flare, retard and ground lock in Figures 16 and 17), the landing occurs. In case it goes outside this region, it enters in one of the states corresponding to hold altitude or go-around. In hold altitude state the UAV is commanded a descent rate of zero, as it is supposed to maintain its altitude until it returns to a safe region. Alternatively, it can assume a go-around logic: the aircraft receives a pitch-up command in order to increase its altitude and move away from a dangerous situation.

As will be demonstrated at the end of next chapter, this state machine logic has great influence on the performance of the landing maneuver. For this reason it is worth to mention (even roughly) its characteristics. A number of additional FSM aspects could be analyzed in more detail while looking carefully into Figure 18, but we prevent to go into further details of the FSM, since it is outside of the scope of this work.

4 Cooperative Control Design

Control design can often be a difficult and iterative task. Knowledge and understanding of the process to be controlled are the key elements for a satisfactory, and ideally optimal performance. In this chapter, the general aspects and design objectives that were taken into consideration during the cooperative controller design phase are discussed.

4.1 Control Loops

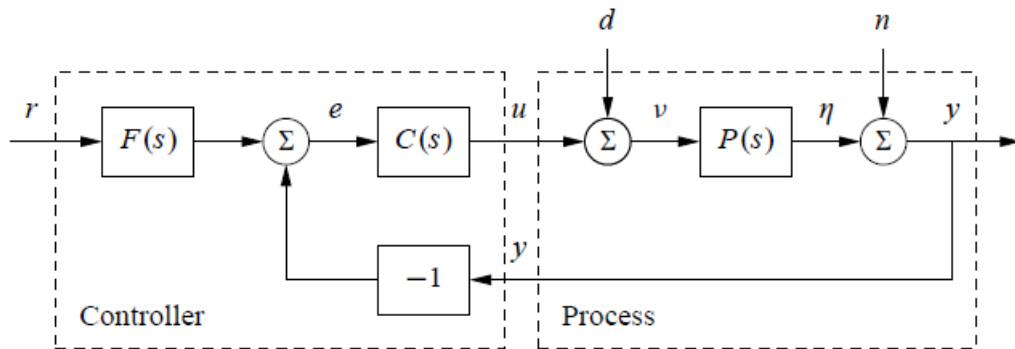


Figure 19 – Basic Feedback Loop.

Source: (ASTRÖM; MURRAY, 2010).

Figure 19 shows a basic control system with unit feedback. The concept of feedback has many interesting properties which go back to Mason (1952) and can be exploited while designing control systems. For example, it can make the system robust to external disturbances and variations in the model.

Another concept very useful in control theory is the so-called feedforward, being a simple and powerful tool to complement feedback. As was explained in Chapter 3, this property has been exploited during the design of the new architecture. Feeding the UAV velocity into the UGV acts like a feedforward term, which helps to damp the relative motion. When a wind gust acts on the aircraft, it immediately causes a change in its velocity, then its position changes accordingly. If there were only the feedback path, this information would have to get accumulated by the aircraft's inertia in order to generate a position error, and only after that being sent to the UGV. With the feedforward term the ground vehicle receives this information much faster.

Two degrees of freedom controllers have the advantage that the reference tracking response can be designed independently from the disturbance rejection and robustness (ASTRÖM; MURRAY, 2010). In the basic feedback loop of Figure 19, the filter $F(s)$ is a feedforward term.

First, the feedback controller $C(s)$ is designed for disturbance rejection. Then, if the system's resulting tracking capability is poor, the feedforward term can be designed to improve it.

An interesting thing to note in the referred feedforward path in Figures 11 and 14 is that we are able to reconstruct the path from the velocity signal back to the input signal of the controller, which is summed to this signal. This could drive us to the conclusion that it is not strictly a feedforward term as we can see in Figure 19. It could be argued that a cascaded control would be a more appropriate term. But cascade usually refers to an outer loop and inner loops with smaller and larger bandwidths respectively (see successive loop closure in Section A.1.1). This is an interesting emerging property from a cooperative control topology. For this reason, in the absence of a better term, we keep feedforward as it embodies the most important properties injected by this informational path.

4.2 Disturbances and Uncertainties

In a non-ideal (but realistic) scenario, there are always disturbances acting on a control system. Setpoint changes, load disturbances, and measurement noise are all external inputs to the control system. It is very important to know what kind of disturbances are acting on the system and which limitation they impose. In this section, we follow the concept of disturbance as any external input to the system according to Aström and Murray (2010).

In most control loops, the setpoint is considered to have a constant value; the same convention is adopted here. In the particular case of having a supervisory controller in the higher level, sending the references Δ_{Xdes} and Δ_{Ydes} to the cooperative control loop, this *setpoint* would change at a specific rate. But this is not addressed in our analysis.

In this application, the typical load disturbances are due to wind gusts and turbulence affecting the airplane dynamics. The effects of the measurement noises were not considered in this analysis. Furthermore, we considered the time delay due to computation and communication through the wireless network, which due to its random nature it is classified as an uncertainty.

4.2.1 Wind Gust

The wind gust profile used in this thesis has the form illustrated in Figure 20. It is modeled as the classical one-minus-cosine, as suggested in (CHALK et al., 1969) according to Equation 4.1.

$$V_{wind} = \begin{cases} 0, & x < 0 \\ \frac{V_m}{s} \left(1 - \cos\left(\frac{\pi x}{d_m}\right) \right), & 0 \leq x \leq d_m \\ V_m, & x > d_m \end{cases} \quad (4.1)$$

where

- $V_m \rightarrow$ gust amplitude.
- $d_m \rightarrow$ gust length.
- $x \rightarrow$ distance traveled.
- $V_{wind} \rightarrow$ resultant wind velocity.

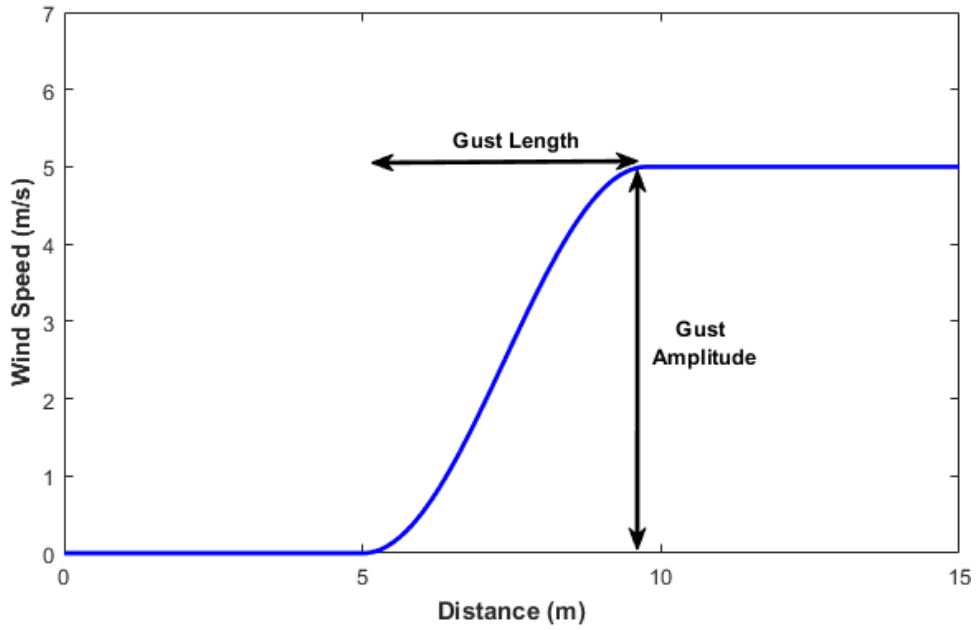


Figure 20 – Discrete Wind Gust Model.

Source: (MATLAB, 2018).

4.2.2 Wind Turbulence

The Dryden wind turbulence model has been used to simulate the turbulence portion of the wind disturbance (LANGELAAN; ALLEY; NEIDHOEFER, 2011). In this model, a band-limited white noise passes through a linear filter defined according to Equations 4.2 and 4.3 for the longitudinal e lateral directions, respectively.

$$H_u(s) = \delta_u \sqrt{\frac{2L_u}{\pi V}} \frac{1}{1 + \frac{L_u}{V}s} \quad (4.2)$$

$$H_v(s) = \delta_v \sqrt{\frac{L_v}{\pi V}} \frac{1 + \frac{\sqrt{3}L_v}{V}s}{\left(1 + \frac{L_v}{V}s\right)^2} \quad (4.3)$$

where

L_u, L_v, L_w represent the turbulence scale length.

$\delta_u, \delta_v, \delta_w$ represent the turbulence intensities.

Typically for low altitudes we have:

$$L_w = h \quad (4.4)$$

$$L_u = L_v = \frac{h}{(0.177 + 0.000823)^{1.2}} \quad (4.5)$$

$$\delta_w = 0.1W_{20} \quad (4.6)$$

$$\frac{\delta_u}{\delta_w} = \frac{\delta_v}{\delta_w} = \frac{1}{(0.177 + 0.000823h)^{0.4}} \quad (4.7)$$

where W_{20} is the wind speed at 20ft (or 6m).

4.2.3 Time Delays

Time delays are uncertainties which can cause serious problems in control systems, since they directly affect its phase margin, decrease its performance, and in a worst-case scenario, could even make the system unstable.

As expected of any control system that relies on a digital communication channel, the cooperative landing problem is also subjected to time delays, and they need to be compensated. Previously, the communication lag was determined from GPS time measurements and the mean value found was 150ms. However, this delay was mainly caused by computational delays, rather than wireless network throughput. In future experiments, the system is expected to exhibit less delay due to an upgrade of the computers on both vehicles.

4.3 Frequency Domain Design

Considering the fundamental problem addressed by this thesis, which is the comparison between different control architectures, an appropriate design procedure is the frequency domain loop shaping. This is related to the fact that two systems can be described as close, meaning they have similar behavior, if their frequency responses are similar (SKOGESTAD; POSTLETHWAITE, 2007).

The frequency domain is closely related to the time domain. For example, a small phase margin indicates the system is poorly damped and a high bandwidth means the system is able to track high frequency commands well.

First, it is important to review some frequency-domain definitions:

- **Gain crossover frequency** → is the point at which the gain curve of the loop transfer function crosses the $0dB$ line in a Bode plot;
- **Bandwidth** → it can be defined as the point where the sensitivity function S crosses $-3dB$ from below;
- **DC gain** → zero frequency magnitude – the gain at very low frequencies;

The design procedure is the following:

1. Analyse time response behavior, determine deficiencies.
2. Plot open-loop system's frequency response.
3. Add controller to change shape of frequency response plot.

In Bode plots, the graph of the controller literally adds to the graph of the plant, since in a log scale a multiplication of transfer functions becomes a summation, that is the reason for using the open-loop for design.

Employing Bode plots with the open-loop frequency responses is a very natural and easy way to design a controller. However, it demands insight from the designer. The magnitude and phase depends on each other. For example, when trying to achieve a certain steady state error by changing the DC gain, or when making the response faster by changing the gain crossover frequency, one may adversely affect the system phase margin and therefore its overshoot (SKOGESTAD; POSTLETHWAITE, 2007). For this reason, this is usually an iterative process.

The design process usually involves adding compensators such as lead, lag and notch filters, adding poles and zeros to the system. However, in this application the controller structure is fixed. The only way to change the shape of the response is by changing the value of the gains and therefore moving the poles and zeros, without adding extra dynamics.

In this part of the design process, the *Control System Tuner* from the MATLAB® Robust Control Toolbox was used to find the initial values for the controller gains. The goal was to make the frequency response of both architectures as similar as possible and pass the corresponding controller gains as initial guesses to the optimization algorithm.

Optimized controllers for different architectures (as described in Åström and Hägglund (1995)) can be compared. Naturally, there are alternative control methods, such as LQG, H -infinity and others, which employ optimization in their structure and could have been used. However, since the architecture currently in use is made of PID loops, it has been decided to keep this structure and optimize the gains numerically. The PID controllers are very useful in practical terms, since its gains are easy to interpret physically which suits for performing manual in-flight tuning if needed.

The points specified in the plots below are the bandwidth of the system ω_B , the gain crossover frequency ω_{gc} and the sensitivity peak, which is a measure of robustness of the system. The ω_B and ω_{gc} points are usually close to each other and they give an indication of the system speed of response. Also, the larger the slope at the crossover frequency ω_{gc} and the greater the bandwidth ω_B , the faster is the system. However, it will also increase overshoot due to a consequent reduction in the phase margin of the system.

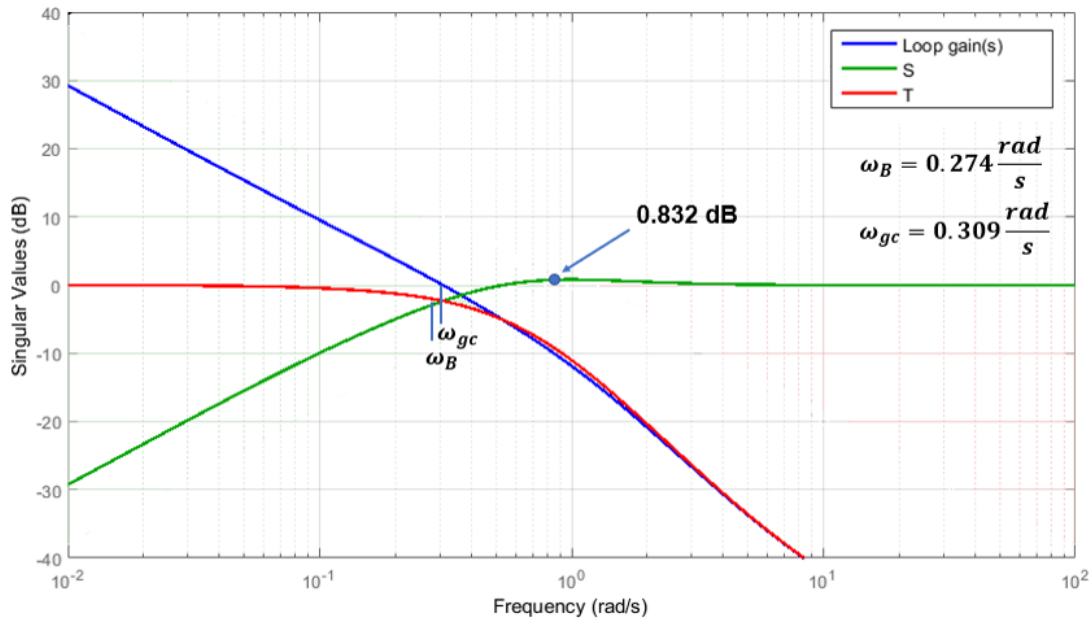


Figure 21 – Loop shape of the longitudinal dynamics of the current architecture before optimization.

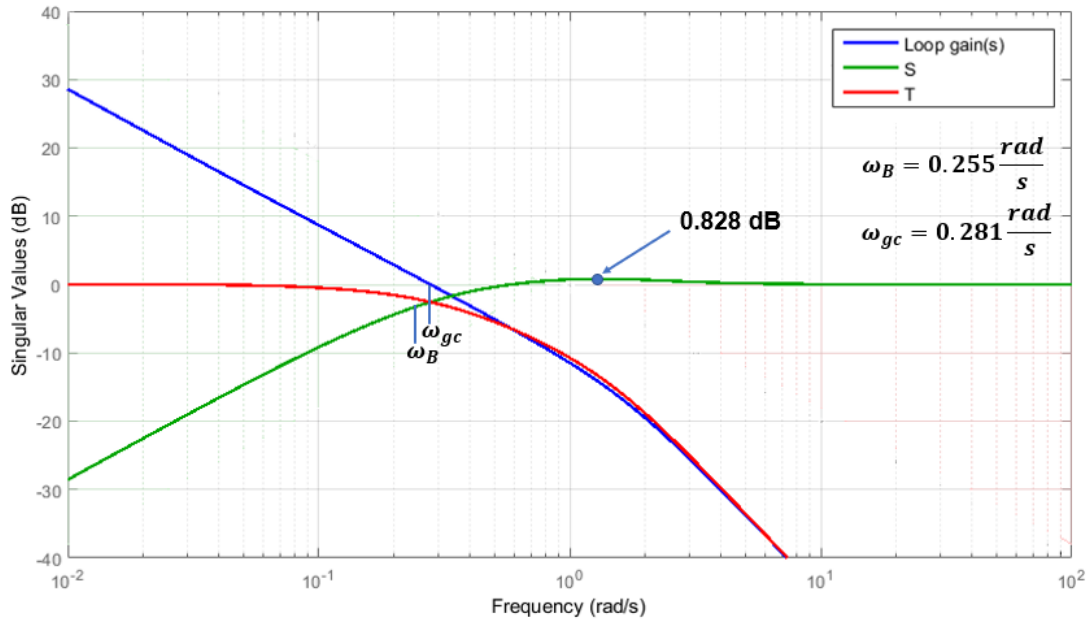


Figure 22 – Loop shape of the longitudinal dynamics of the proposed architecture before optimization.

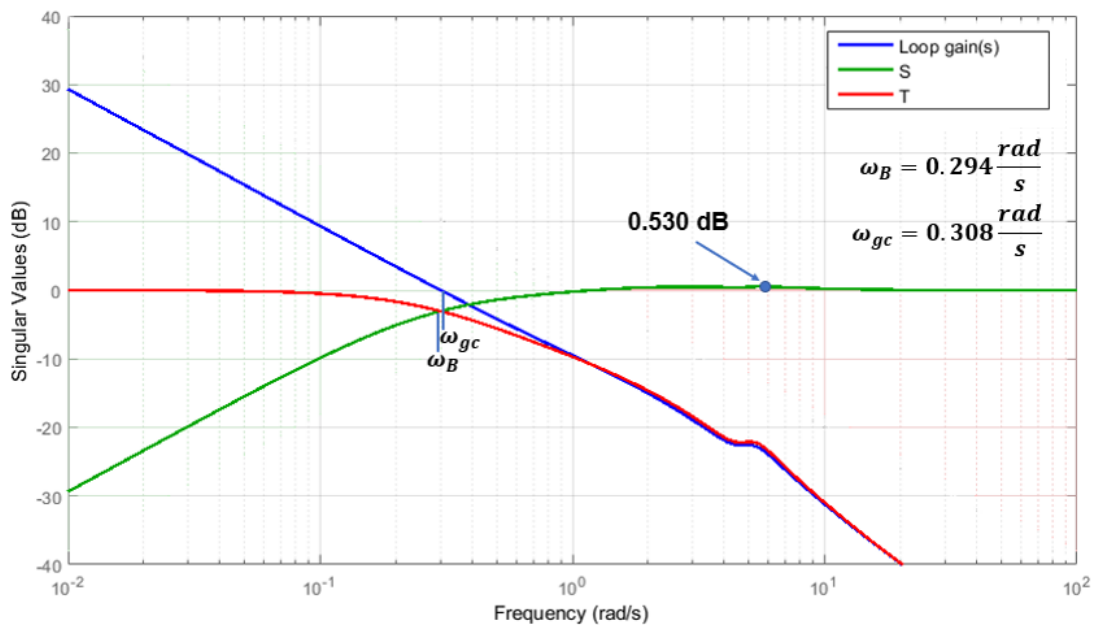


Figure 23 – Loop shape of the lateral dynamics of the current architecture before optimization.

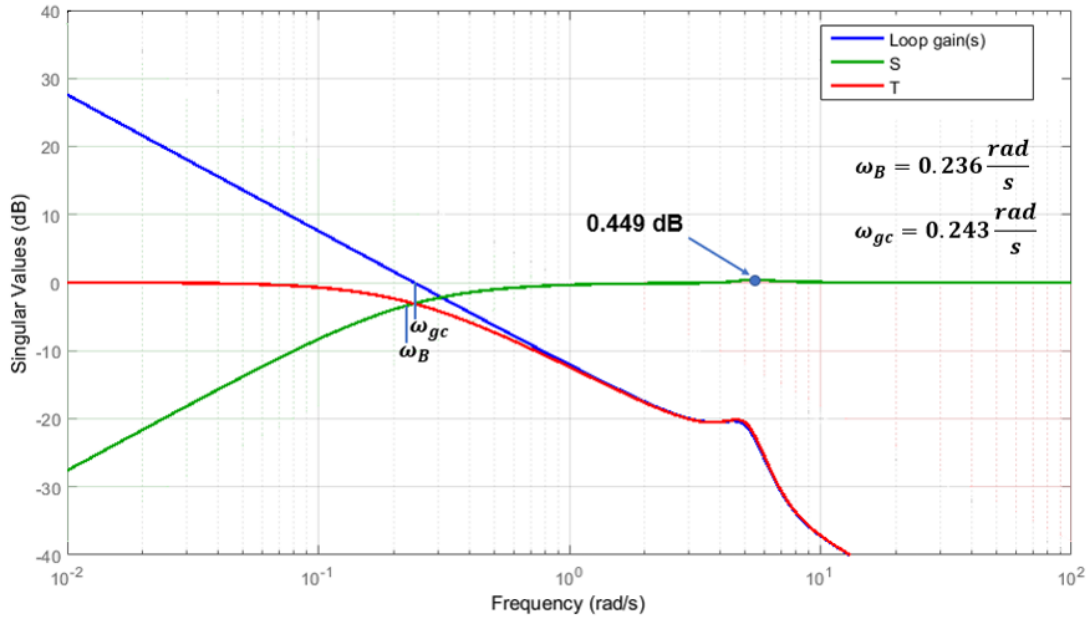


Figure 24 – Loop shape of the lateral dynamics of the proposed architecture before optimization.

Figures 21 and 22 show the frequency response for the longitudinal dynamics and Figures 23 and 24 represent the lateral dynamics. It can be noticed how close the responses are. This means that with this configuration and corresponding gain values, the two architectures have a similar performance.

At this point we establish both architectures for lateral and longitudinal as being dynamically similar, where the measures of similarity are: gain crossover frequency, bandwidth, and sensitivity peak. This effort is assumed as an argument in favour of a fair starting point for numerical optimization of current and proposed architectures.

4.4 PID Controller

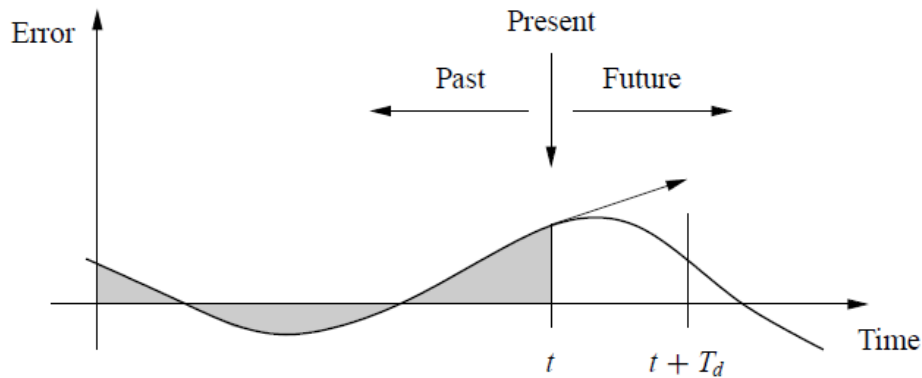


Figure 25 – Actions of a PID controller. At time t , the proportional term depends on the instantaneous value of the error. The integral portion of the feedback controller is based on the integral of the error up to time t (shaded region). The derivative term provides an estimate of the growth or decay of the error over time by looking at the rate of change of the error. T_d represents the approximate amount of time the error is projected forward.

Source: (ÅSTRÖM; MURRAY, 2010).

The PID controller is certainly the most widely used control strategy in industry. This great success comes from its simplicity and versatility. In a basic feedback loop it can eliminate steady state error through integral action; it can predict the future through its derivative term, and most importantly, it is usually model independent, which makes it robust to model uncertainties.

PID control is often implemented in conjunction with logic of state machines, enabling a smooth switching between the different modes in an application. It is the main component of simple functions blocks that together form a complicated automation system, such as the cooperative architecture. Many sophisticated control strategies, for instance, model predictive control (MPC), sometimes make use of a PID controller in the lower level, with the multivariable controller giving the setpoints to the lower level PID loop (ÅSTRÖM; HÄGGLUND, 1995).

With these facts in mind, and due to the previous experience of the group in designing this type of controller, the new architecture of the cooperative landing is made up of a combination of PID controllers.

4.4.1 Controller Tuning

A similar procedure to that described in Section A.2.3 of Appendix A has been employed to get to an optimal set of gains for the cooperative controllers. The initial condition was defined with both vehicles aligned in the longitudinal and lateral axis. A disturbance is then applied into the system; therefore, within this setup it is not suitable to use the rise time information. The rise

time term was then removed and a term due to the undershoot has been added. The resulting cost function has the form of Equation 4.8.

$$f(OS_{max}, US_{max}, |\dot{u}|, ITAE) = w_1 OS_{max} + w_2 US_{max} + w_3 |\dot{u}| + w_4 ITAE \quad (4.8)$$

where OS_{max} is the maximum overshoot, US_{max} is the maximum undershoot, \dot{u} is the derivative of the control command and ITAE is a well-known performance index based on the integral of the time multiplied by the absolute error.

The basic idea is that both closed-loop systems started with a similar frequency response, which means they were equivalent systems prior to the optimization. Thus, the optimization algorithm described in Appendix A with the cost function of Equation 4.8 is used to explore the full potential of the architectures. This is a fair way to compare the architectures, since the gains were selected with the same criteria in all phases of the design process. Any possible difference in performance is due to the architecture structure.

Also, in order to evaluate the effectiveness of the optimization method for gain tuning, we compare the stability margins of the longitudinal dynamics before and after the optimization in Chapter 6.

Additionally, the final value of the cost function after optimization can be used as a metric for comparison between the architectures. As shown in Figures 26 and 27, the proposed architecture exhibited a value more than two times smaller for the final cost of the objective function. This means a better performance in terms of disturbance rejection. It was possible to make this claim because the requirements and weights of the cost functions were the same for both. It is important to mention that this tuning procedure has been made for disturbance rejection and, as a result, the bandwidth of the system is not so high. It is expected that tuning for setpoint tracking yields a set of gains which would make the system faster.

Tables 2 and 3 show the final values of the gains. Note that the derivative terms are expressed as velocity feedbacks (proportional to the error in velocity, $K_{p\dot{X}}$ and $K_{p\dot{Y}}$), as previously mentioned.

Table 2 – Values tuned with the optimization method for the current architecture.

	Gain	Optimal value
<i>Longitudinal</i>	K_{pXG}	-0.3388
	K_{iXG}	0
	$K_{p\dot{X}G}$	0.0003787
<i>Lateral</i>	K_{pYA}	0.88948
	K_{iYA}	21.5695
	$K_{p\dot{Y}A}$	-0.20062

Table 3 – Values tuned with the optimization method for the proposed architecture.

	Gain	Optimal value
<i>Longitudinal</i>	K_{pXG}	-0.3844
	K_{iXG}	0
	$K_{p\ddot{X}G}$	0.2701
<i>Lateral</i>	K_{pYG}	-0.7907
	K_{iYG}	-0.002971
	$K_{p\ddot{Y}G}$	1.6877
<i>Longitudinal</i>	K_{pXA}	0.2366
	K_{iXA}	0.09508
	$K_{p\ddot{X}A}$	-0.7221
<i>Lateral</i>	K_{pYA}	1.7616
	K_{iYA}	0.00357
	$K_{p\ddot{Y}A}$	-0.0589

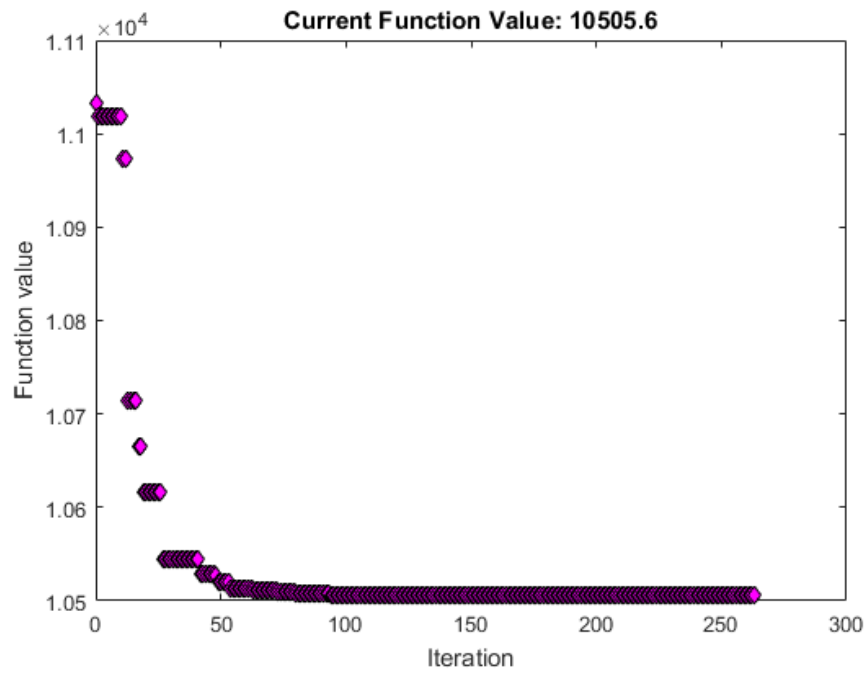


Figure 26 – Cost function versus number of iterations for current architecture.

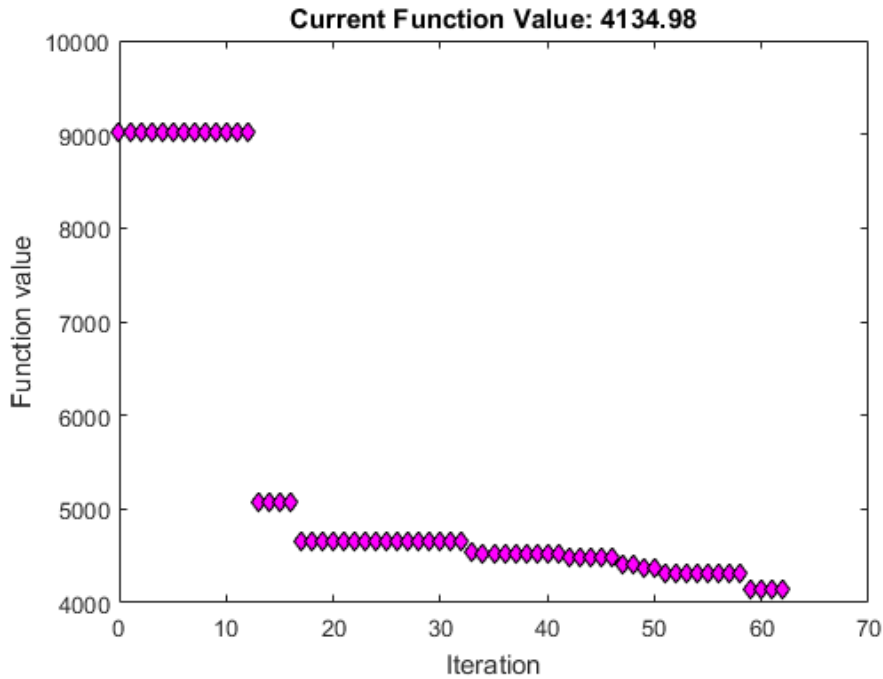


Figure 27 – Cost function versus number of iterations for proposed architecture.

4.5 Stability Assessment

In every application of cooperative control a primary concern is stability. Since the vehicles are connected through a digital system, they are susceptible to time delays inherent to computation and network latency. It undermines the performance of the system and, in a worst-case-scenario, can make the system unstable. For that reason, time delays need to be accounted for during the control design.

Analysing the poles of the closed loop systems in Figures 28 and 29 and the corresponding values below the respective figures, it can be seen that they are all located in the left half plane or at the origin for both architectures. Hence, the system is marginally stable without time delay. It can also be noticed that the longitudinal dynamics of the proposed architecture is much more complex, with a greater number of poles and zeros, as expected.

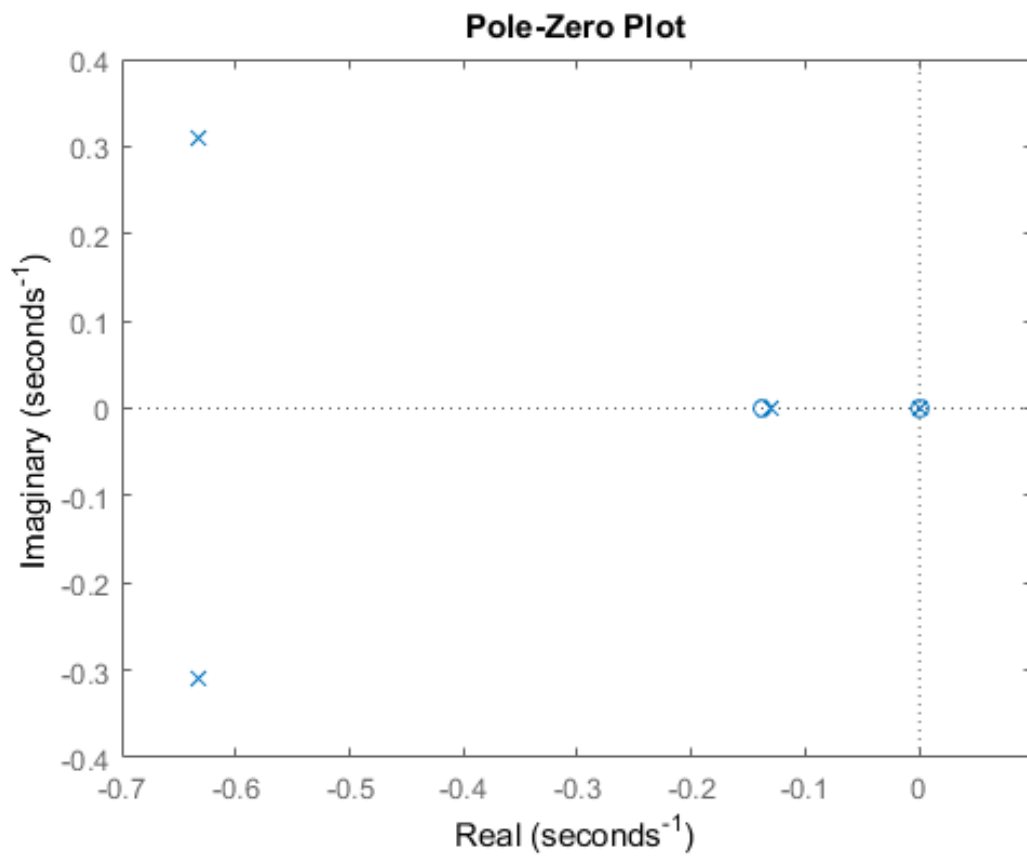


Figure 28 – Pole-Zero plot for the transfer function $\frac{\Delta x}{\Delta x_{des}}$ of the current architecture.

-0.6334 + 0.3084i
 -0.6334 - 0.3084i
 -0.1302 + 0.0000i
 -0.0000 + 0.0000i

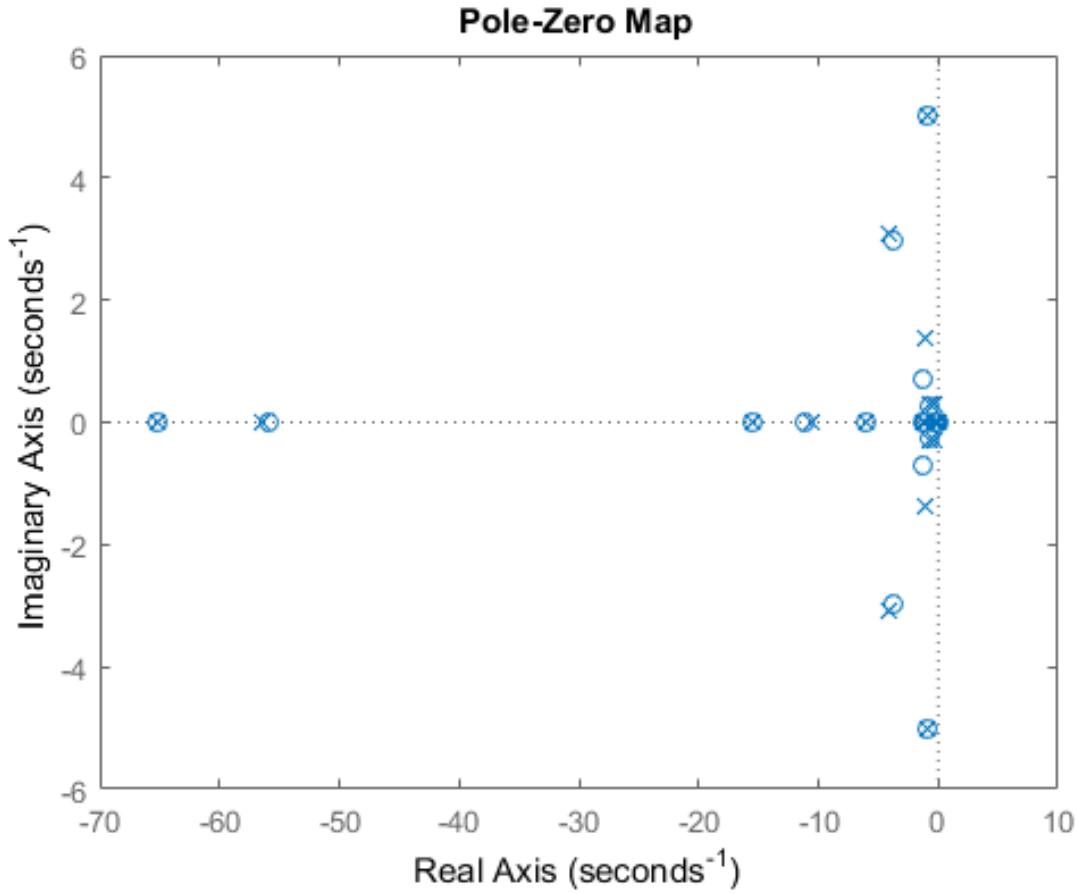


Figure 29 – Pole-Zero plot for the transfer function $\frac{\Delta x}{\Delta x_{des}}$ of the proposed architecture.

$-56.3862 + 0.0000i$	$-1.1109 - 1.3739i$	$-0.1809 + 0.0000i$
$-10.4833 + 0.0000i$	$-0.7417 + 0.3027i$	$-0.1749 + 0.0000i$
$-4.1865 + 3.0959i$	$-0.7417 - 0.3027i$	$-0.1073 + 0.0000i$
$-4.1865 - 3.0959i$	$-0.3873 + 0.3016i$	$0.0000 + 0.0000i$
$-1.1109 + 1.3739i$	$-0.3873 - 0.3016i$	$0.0000 + 0.0000i$

In order to reach actual position synchronization, a simple estimation of the form of Equation 4.9 was used. As the position data is exchanged, a time delay is introduced; the arriving data is old, therefore the old position must be corrected in order to achieve position synchronization. The old position data is then linearly extrapolated using the previous velocity data which was measured at the same time and also comes through the communication channel. Both position and velocity measurements are obtained through GPS sensors and thus contain a synchronized GPS-timestamp. This makes it possible to estimate the time delay between the current local measurement and the old data received through the network using a simple linear estimation.

$$s_{t+\Delta t} = s_t + \Delta t V_t \quad (4.9)$$

To guarantee the stability of the overall system even in the presence of time delays, a strategy from the field of tele-operation called Time-Domain-Passivity-Approach (TDPA) has been implemented by the group (RYU; ARTIGAS; PREUSCHE, 2010). This method uses a scheme of a Passivity-Observer-Passivity-Controller (PO-PC) to guarantee the system passivity and thus stability. It monitors the energy flow into and out of the communication channel and reduces the control action if an increase of energy is detected. The control commands sent to the vehicles are dynamically reduced depending on the amount of excess energy.

4.6 Control Validation

As a last step in this chapter, the controller design was verified in simulation using the nonlinear models introduced in Chapter 2. The results of the simulated landings procedures were also used as a means of performance comparison between the architectures.

4.6.1 Simulation Tests

Following the description of the modifications introduced with the proposed architecture in Chapter 3 and the procedure described in this chapter for controller design and tuning, the next step is to test the validity of the design within a realistic simulation environment. Naturally, for a complete validation of the control scheme it is necessary to test on the real hardware, but due to time constraints these experiments still have to be performed and therefore were not included in this thesis.

4.6.2 Landing

A cooperative landing procedure, as previously stated in Chapter 3, is composed of 3 parts: longitudinal and lateral alignment, and vertical descent.

Some time-domain requirements such as maximum overshoot, rise time, settling time and steady-state error were used to judge the performance of both architectures. Figure 30 shows how these values are defined for a unit step input.

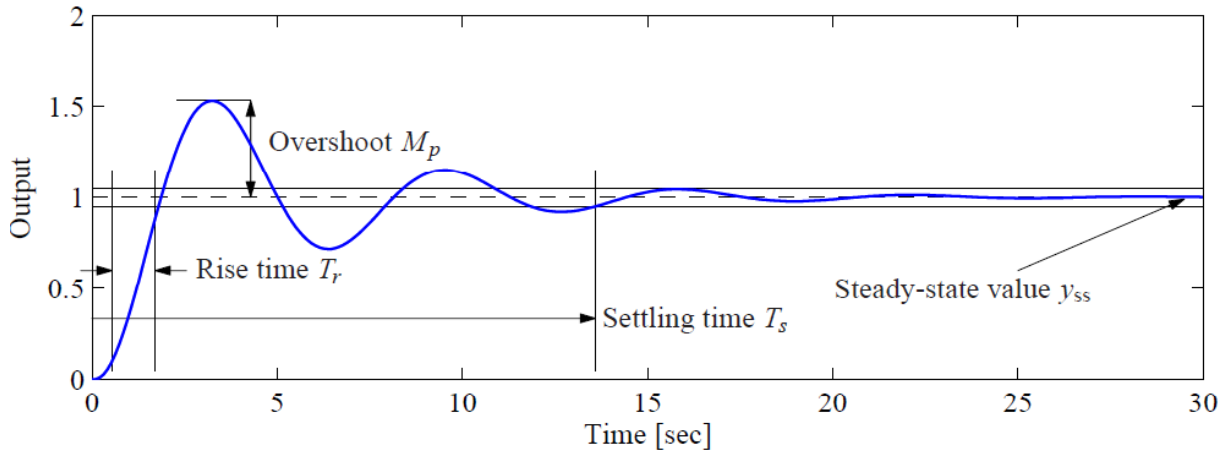


Figure 30 – Time-Domain Requirements.

Source: (ASTRÖM; MURRAY, 2010).

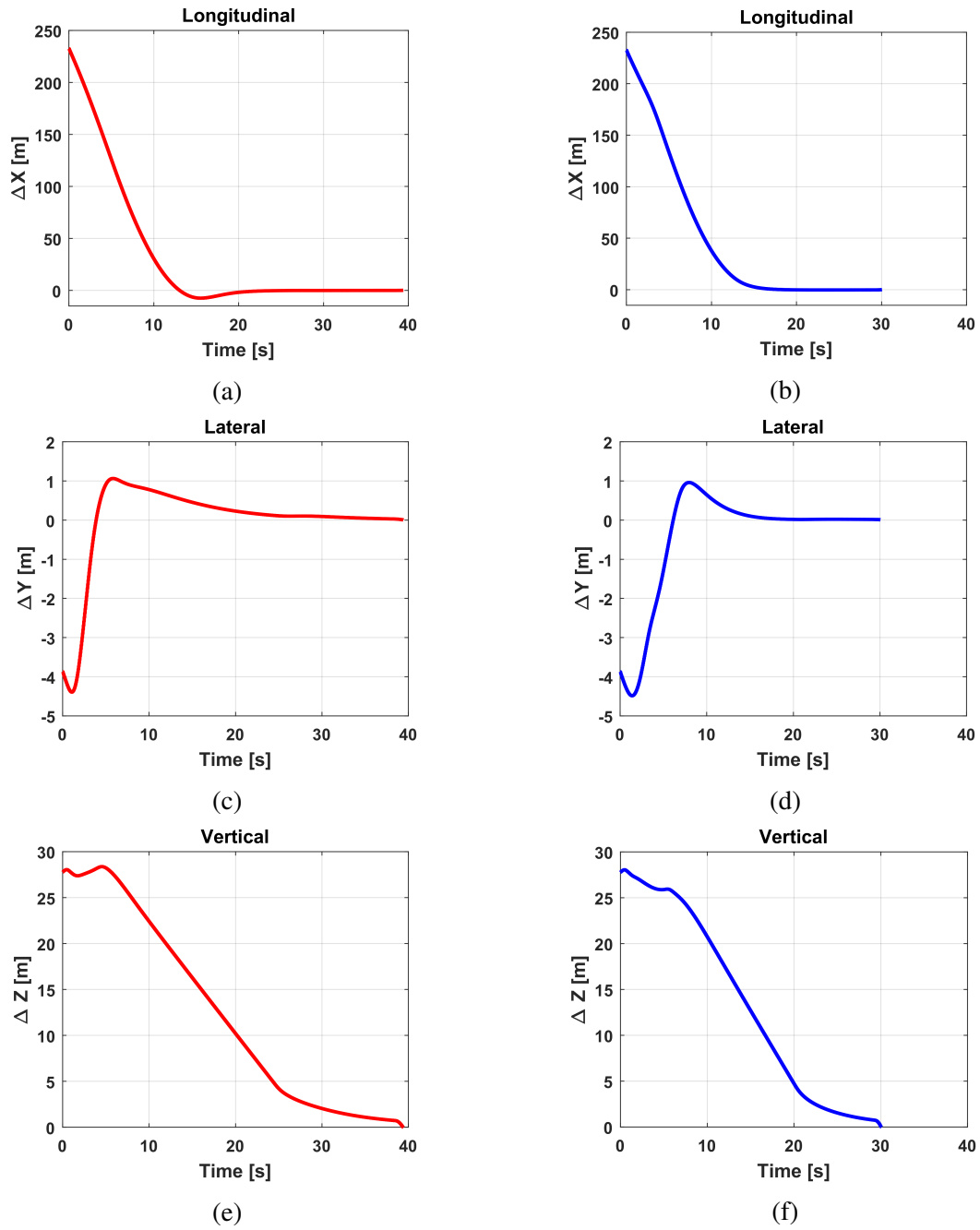


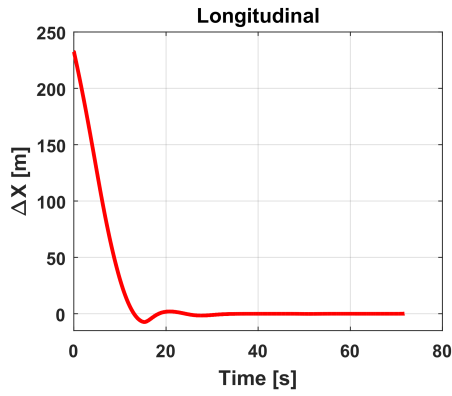
Figure 31 – Landing maneuver without wind gust. The red plots on the left belong to the current architecture while the blue plots on the right are from the proposed architecture.

Table 4 – Landing maneuver with the current architecture without disturbance.

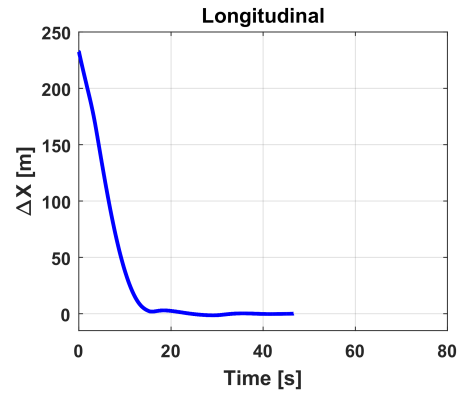
Performance Measure	Longitudinal	Lateral	Vertical
Rise Time (t_r)	13.05s	3.9s	-
Settling Time (t_s)	24.39s	20.99s	-
Overshoot (OS_{max})	7.34m	1.06m	-
Undershoot (US_{max})	0.15m	0m	-
Touchdown Time	-	-	39.40s

Table 5 – Landing maneuver with the proposed architecture without disturbance.

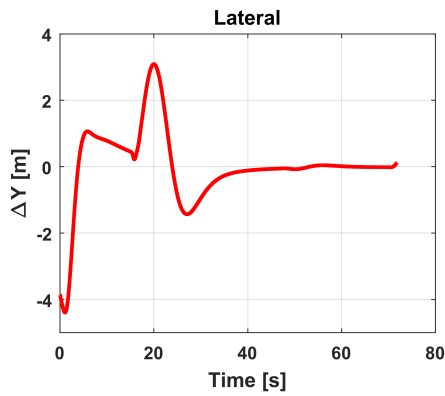
Performance Measure	Longitudinal	Lateral	Vertical
Rise Time (t_r)	18.80s	6.15s	-
Settling Time (t_s)	24.39s	13.34s	-
Overshoot (OS_{max})	0.20m	0.96m	-
Undershoot (US_{max})	0.02m	0m	-
Touchdown Time	-	-	30.05s



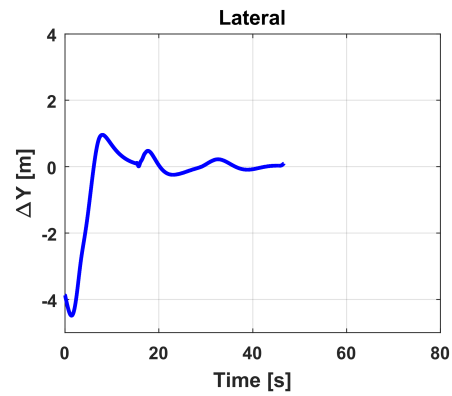
(a)



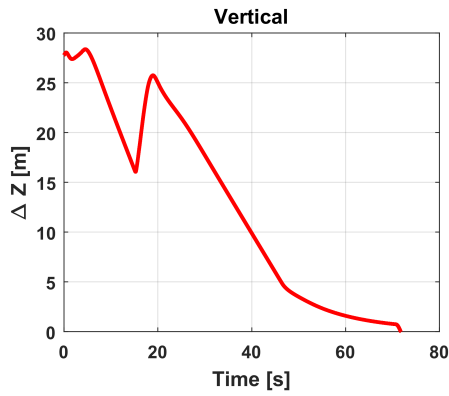
(b)



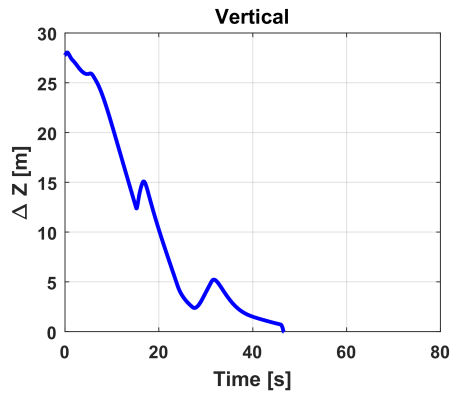
(c)



(d)



(e)



(f)

Figure 32 – Landing maneuver with wind gust. The red plots on the left belong to the current architecture while the blue plots on the right are from the proposed architecture.

Table 6 – Landing maneuver with the current architecture in the presence of disturbances.

Performance Measure	Longitudinal	Lateral	Vertical
Rise Time (t_r)	13.05s	3.95s	-
Settling Time (t_s)	33.59s	36.44s	-
Overshoot (OS_{max})	7.39m	3.09m	-
Undershoot (US_{max})	2.05m	1.43m	-
Touchdown Time	-	-	71.75s

Table 7 – Landing maneuver with the proposed architecture in the presence of disturbances.

Performance Measure	Longitudinal	Lateral	Vertical
Rise Time (t_r)	24.10s	6.20s	-
Settling Time (t_s)	42.24s	33.54s	-
Overshoot (OS_{max})	1.38m	0.96m	-
Undershoot (US_{max})	0.34m	0.24m	-
Touchdown Time	-	-	46.60s

As stated in Section 3.5.3 the landing experiments were performed using the same vertical control strategy, therefore the differences in performance are due to the longitudinal and lateral motions only.

Another important fact worth of mentioning is that the optimal gains for the cooperative architectures found in Section 4.4.1 could not be used directly in the nonlinear models. Thus, it was necessary to make an optimization with the nonlinear models as well. This is one of the advantages of the optimization method explained in Section A.2.3, since it only checks the output signal in order to calculate a cost and affects the system by changing the cooperative gains. Thus, it is model independent and it can be used in both linear or nonlinear simulations (HAROLD; RANDAL, 2011).

Analysing Figure 31 and the corresponding Tables 4 and 5, it is evident that there is not a great difference in performance between the two architectures considering the complete landing maneuver. Apart from the current architecture presenting more overshoot in its motion and the proposed structure being relatively faster. Still, these alone do not justify the use of the proposed structure, since the previous design has some advantages, for instance, it is easier to implement.

Nevertheless, Figure 32 and Tables 6 and 7 show a different situation, when there are disturbances acting on the system. It can be seen that the new architecture is less affected by them. This becomes evident when looking the overshoot values. In the longitudinal motion, the current architecture had 7.39m of overshoot, while the proposed scheme had only 1.38m. This behavior is also noticeable in the lateral motion where we have 3.09m for the current scheme and only 0.96m for the proposed architecture.

This behavior directly affects the time required for landing. As can be observed in the last line of Tables 6 and 7, the time until touchdown for the current architecture was 71.75s while for the proposed architecture was just 46.6s.

This result is a consequence of the state machine described in Section 3.6; because of the oscillation caused by the disturbance, it takes longer for the aircraft to reach the state of landing, remaining in the state of descent for a longer period of time. Moreover, in plot *f* of Figure 32, between the instants $27s$ and $31s$ the airplane enters in *low go around to retry* mode in the state machine (see Figure 17) due to the position error. Another interesting aspect in this response is the abrupt change in altitude illustrated in plots *e* and *f* of Figure 32 between the instants $15s$ and $20s$. This response is inherent to the aircraft longitudinal controller (TECS), briefly introduced in Section A.1.4 of Apendice A. The wind gust causes an increase in the airplane velocity, the flight controller then enters in speed priority mode in which it cancels the throttle command and control the speed with the elevator only. These details of the aircraft flight controller are better described in Balmer (2015).

The wind gust applied to the system has the shape as described in Section 4.2.1 and was activated at $15s$ into the landing maneuver, coming from the North-East direction with a magnitude of $10m/s$, which is considered a moderate disturbance.

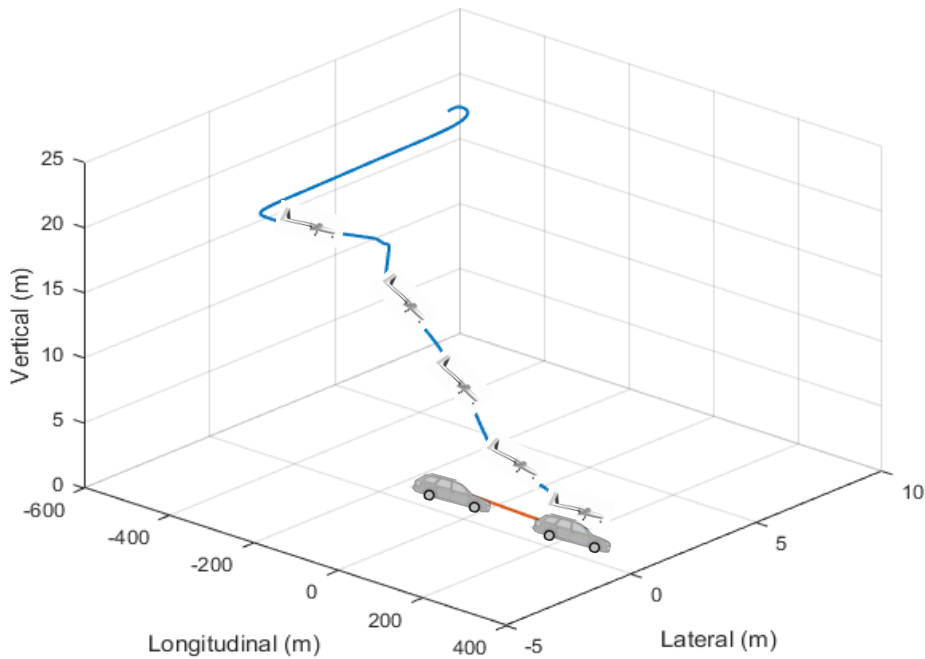


Figure 33 – 3D representation of a landing maneuver.

Figure 33 illustrates a landing maneuver in three dimensions. The procedure described in detail in Section 1.3.1 can be recognized.

5 Sensitivity Analysis

This chapter explains the methodology used for the sensitivity analysis of the control architectures from Chapter 3. First, the time-domain approach based on Monte Carlo simulations is explained. Thereafter, the frequency-domain analysis is described.

5.1 Definition

Sensitivity analysis is the study of how variations in the parameters or inputs affect the output behavior of the model (SALTELLI et al., 2008). It brings many advantages to the system, for example, it increases reliability and robustness, it helps to reduce costs, and it improves its performance.

5.2 Sensitivity in the Time Domain

5.2.1 Monte Carlo Simulation

A useful and practical way to assess the robustness of a control system is through Monte Carlo experiments. In the real world, precise measurements or observation of the parameters in a process may be difficult, or it is possible that the values drift over time as the components age. Also, some inputs may be probabilistic in nature, as are the wind gusts. The basic idea behind the Monte Carlo method is that by repeatedly sampling from known probability distributions of a random variable, one can get to an empirical probability distribution and from that find the system bounds and trends (HAROLD; RANDAL, 2011). This process is accomplished in simulation by using random number generators and known probability distributions. For this analysis, the MATLAB[®] random number generator was used.

Monte Carlo simulations are particularly appropriate when the process underlying probabilities are known but the results are difficult to determine. That is the main reason why it is vastly used in aeronautics and many other branches of engineering and physics. For many systems, it is possible to write down the laws of physics while modeling it, but it might not be possible to analytically solve the given equations for problems of interest.

In the simulation run cases for the Monte Carlo experiments performed in this thesis, the initial conditions are different from that of conventional control problems, since the references for relative position of the vehicles (ΔX_{des} and ΔY_{des}) have been set to zero and the system must return to this condition after the disturbance has settled down. This condition is called a regulator problem (SKOGESTAD; POSTLETHWAITE, 2007).

The outputs of interest in a Monte Carlo simulation vary according to the simulation setup and initial conditions. Using the linear models derived in Chapter 2, and considering they were linearized at specific points, the simulation starts with the vehicles at the same speed (21m/s) and position ($x = 100\text{m}$, $y = 0\text{m}$) in the horizontal plane. Then, a gust disturbance is added to the system. Therefore, in this case, the outputs of interest are the maximum overshoot and undershoot, the ITAE criteria, the mean absolute error in steady state, and the settling time after a gust.

The random variables used as inputs to the model were the wind turbulence, implemented as a white noise passing through a Dryden filter as described in Section 4.2.2. This turbulence is added to the output of the discrete wind gust model illustrated in Figure 20. Finally, a normal distribution for the time delay has been added into the system. Figure 34 shows a typical output signal from a Monte Carlo experiment, the ripples in the signal are due to these random inputs.

First, experiments with only the wind gust disturbance have been made, in order to divide the problem. Then, the time delay has been activated in order to show the coupling effect between the random inputs, as it is expected to happen in the real experiment.

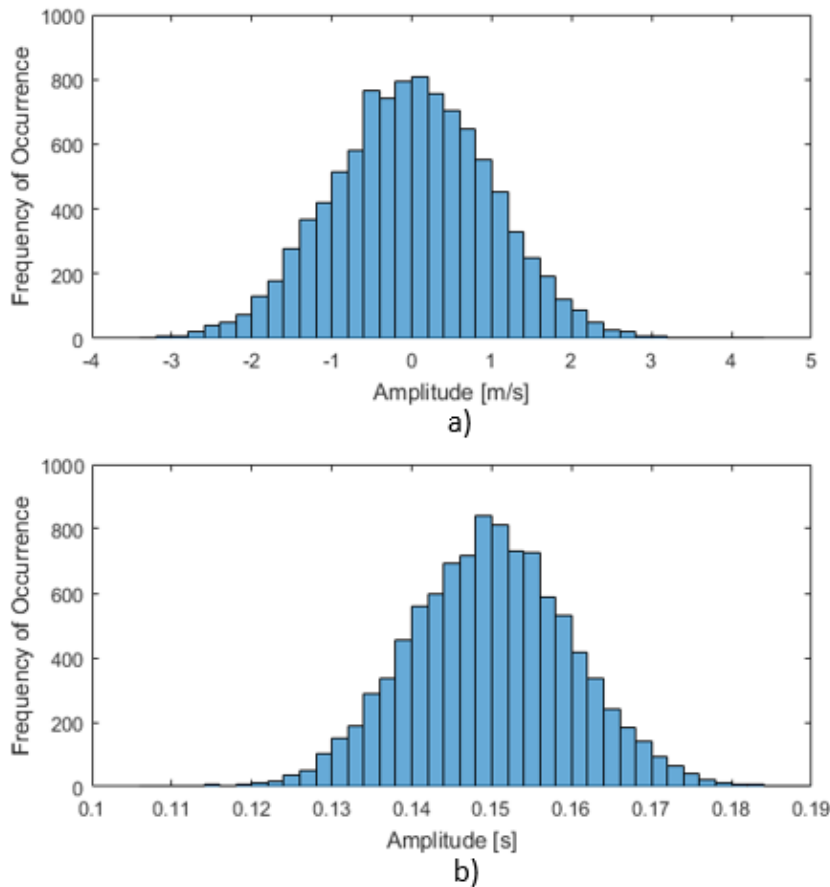


Figure 34 – Distributions of the random inputs. a) shows the turbulence normal distribution, with mean zero and standard deviation of 1; b) represents the time delay distribution, with mean 0.150 and standard deviation of 0.01.

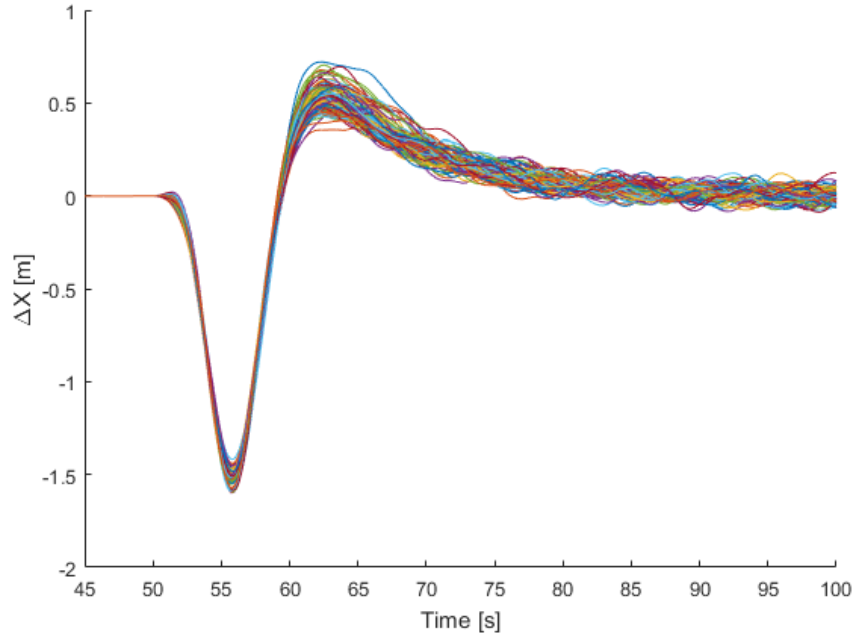


Figure 35 – A typical signal output from a Monte Carlo experiment. The variations in the curves are caused by the random inputs: wind turbulence and time delay. This plot is from one of the experiments with the current architecture.

An example of a typical result from a Monte Carlo simulation is shown in Table 8 and illustrated in Figure 36. The empirical probability density function $\hat{f}(u)$ is obtained by connecting the points of the first and third columns and rescaling the ordinate values f_i . Equation 5.1 is then used to make the area under the curve equal to 1 (HAROLD; RANDAL, 2011). The probability of Overshoot (OS_{max}) falling in a certain interval is the area under $\hat{f}(u)$ in that interval.

There are some methods to select specific parameters in the distribution. For example, the normalization procedure can be performed according to Equation 5.1 and the number of bins in the histogram is usually chosen according to Equation 5.2. While for the number of iterations, as the Monte Carlo method is based on random numbers, it follows the law of large numbers: The probability grows with the number of iterations. The experiments in this thesis were performed with 1000 evaluations each, which has been shown to be accurate enough.

Table 8 – Monte Carlo simulation results for overshoot in the longitudinal direction.

Interval (m)	Center of Interval (m)	Frequency of Occurrence	Normalized Frequency of Occurrence
(0, 0.2590)	0.1295	1	0.1333
(0.2590, 0.2665)	0.2628	1	0.1333
(0.2665, 0.2740)	0.2703	1	0.1333
(0.2740, 0.2815)	0.2778	1	0.1333
(0.2815, 0.2890)	0.2853	4	0.5333
(0.2965, 0.3040)	0.3003	10	1.3333
(0.3040, 0.3115)	0.3078	30	4.0000
(0.3115, 0.3190)	0.3153	42	5.6000
(0.3190, 0.3265)	0.3228	90	12.0000
(0.3265, 0.3340)	0.3303	91	12.1333
(0.3340, 0.3415)	0.3378	124	16.5333
(0.3415, 0.3490)	0.3453	145	19.3333
(0.3490, 0.3565)	0.3528	139	18.5333
(0.3565, 0.3640)	0.3603	114	15.2000
(0.3640, 0.3715)	0.3678	86	11.4667
(0.3715, 0.3790)	0.3753	53	7.0667
(0.3790, 0.3865)	0.3828	35	4.6667
(0.3865, 0.3940)	0.3903	19	2.5333
(0.3940, 0.4015)	0.3978	8	1.0667
(0.4015, 0.4090)	0.4053	6	0.8000

$$f_i = \frac{n_i}{(\text{Number of trials} \times \text{width of interval})} \quad (5.1)$$

$$N^{\circ} \text{ of Bins} = \sqrt{NUMEL(X)} \quad (5.2)$$

where n_i is the frequency of occurrence before normalization (third column in Table 8), *number of trials* is the total number of observations in the Monte Carlo experiment, the *width of interval* is the horizontal separation within each interval, $NUMEL(X)$ is the number of elements in the vector and *Number of Bins* is the number of rectangles in the histogram.

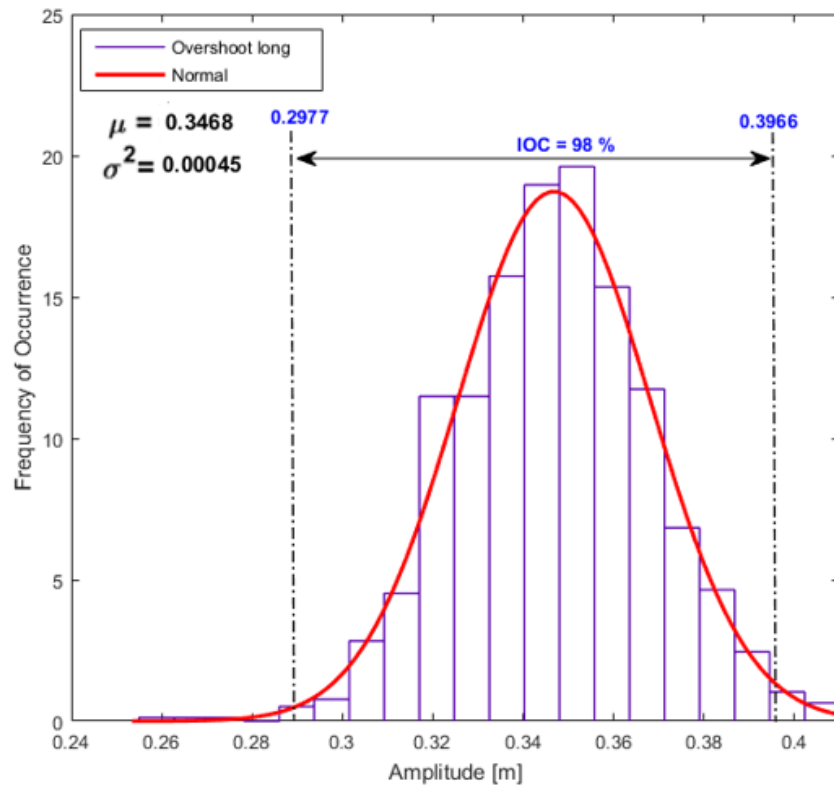


Figure 36 – A typical empirical Probability Density Function. This histogram corresponds to the data in Table 8 and is for the overshoot in the longitudinal direction, which exhibits a normal distribution. Where μ is the mean, σ^2 is the variance, and IOC is the interval of confidence.

5.3 Sensitivity in the Frequency Domain

5.3.1 Feedback Fundamentals

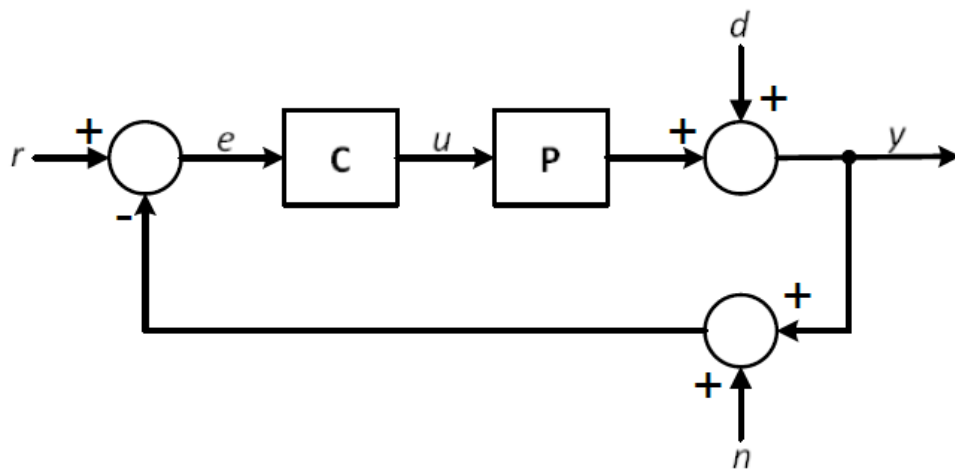


Figure 37 – Feedback loop.

Considering the basic feedback loop in Figure 37, where r is the reference, d is the disturbance, n is the noise input, e is the error, u is the control command and y is the output of the system, it follows that

$$\frac{e}{r} = \frac{1}{(1 + PC)} = S = \text{Sensitivity Function} \quad (5.3)$$

$$\frac{y}{r} = \frac{PC}{(1 + PC)} = T = \text{Complementary Sensitivity Function} \quad (5.4)$$

$$y = (1 + PC)^{-1}PCr + (1 + PC)^{-1}d - (1 + PC)^{-1}PCn \quad (5.5)$$

$$e = r - y = Sr - Sd + Tn \quad (5.6)$$

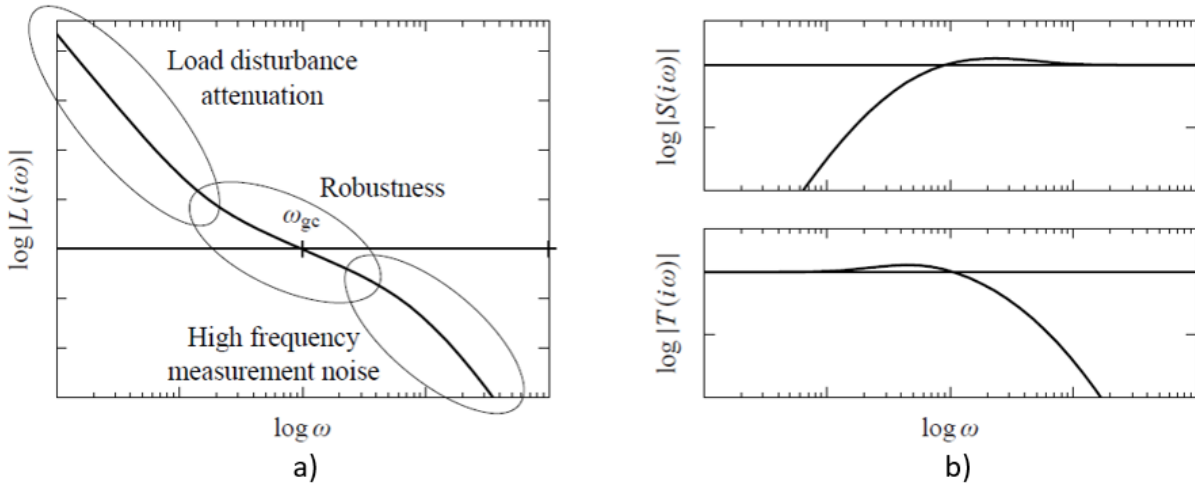


Figure 38 – Gain and sensitivity functions curves for a typical loop transfer function. In a) is shown the gain curve of the loop transfer function ($L = PC$); b) shows the sensitivity (S) and complementary sensitivity (T) functions.

Source: (ASTRÖM; MURRAY, 2010).

Some important parameters in the frequency response plots are the gain crossover frequency ω_{gc} , the slope of the curve at crossover, the sensitivity peak, and the high frequency roll-off (reduction of amplification), which provide an indication of the robustness of closed-loop systems.

For typical applications, the following holds:

- The reference signal $r(t)$ is a low frequency signal with its main frequency content below ω_{gc} ;

- The noise signal $n(t)$ is a high frequency signal with its main frequency content above ω_{gc} ;
- The input disturbance signal $d(t)$ is a low frequency signal with its main frequency content below ω_{gc} ;

The goal is to design a controller $C(s)$ so that the following specifications are satisfied:

- Good Reference Tracking;
- Noise Attenuation;
- Disturbance Rejection;

The idea is to design the controller C to make the sensitivity function S small at low frequencies, to reject disturbances. Also, it is required to make the complementary sensitivity T high at low frequencies for reference tracking and low at high frequency for noise attenuation. Usually, a trade-off between these requirements is necessary. For example, to make the system follow references, the complementary sensitivity needs to be made higher, but this can also amplify noises. An advantage is that these inputs act on different frequency bands. While the reference and the disturbance are low frequency signals, the noise is high frequency. Hence, it is usually possible to accomplish both goals simultaneously.

5.3.2 Design Considerations

$$S + T = 1 \quad (5.7)$$

As mentioned in the last section, it is difficult to reduce the sensitivity without adversely affecting the complementary sensitivity, this is due to the postulate 5.7, which states that the values of the sensitivity and complementary sensitivity add up to 1 across all frequencies.

If the goal is to track faster references and disturbances, the point where the sensitivity function of Figure 38 crosses the $-3dB$ from below needs to be moved to the right, increasing the bandwidth of the system. However, a consequence of getting more speed is that more high frequency signals, such as noise, will enter the system.

Some practical consequences of this have been demonstrated by Stein (2003) in the controller design for the unstable experimental aircraft X-29. Hence, it is extremely important for the designer to be aware of these fundamental limitations.

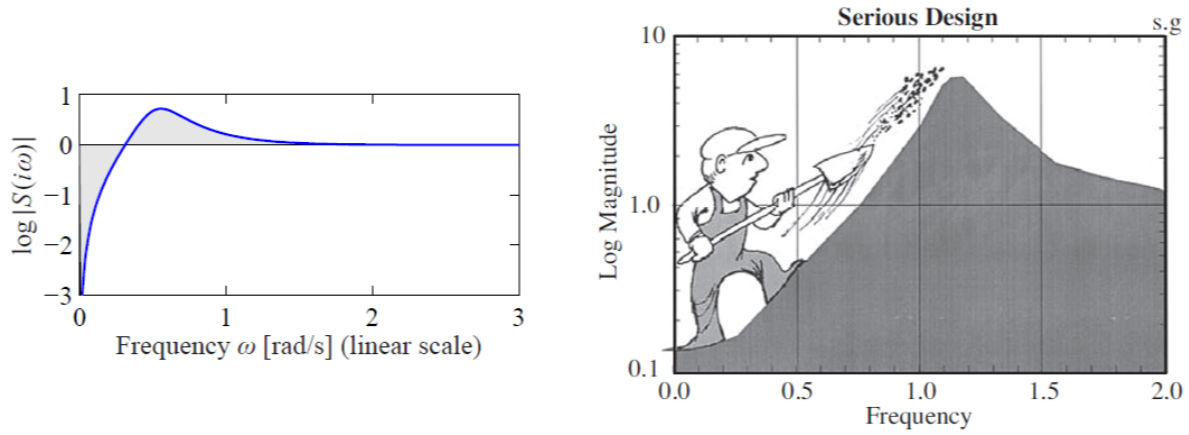


Figure 39 – The trade-off in the design as seen from the sensitivity plot. To make the system better in one frequency range, it is necessary to make it worse at other frequencies. This is sometimes referred to as the waterbed effect.

Source: (STEIN, 2003).

$$M_s = \max_{0 \leq \omega < \infty} |S(j\omega)| = \max_{0 \leq \omega < \infty} \left| \frac{1}{1 + G(j\omega)C(j\omega)} \right| \quad (5.8)$$

Equation 5.8 shows how the maximum value of the sensitivity function is calculated and Figure 40 illustrates what this value represents in terms of the Nyquist plot, as it is the minimum distance from the critical -1 point.

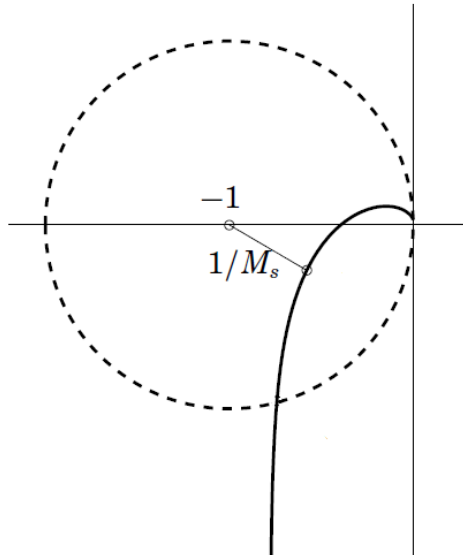


Figure 40 – Nyquist curve of loop transfer function showing the graphical interpretation of maximum sensitivity.

Source: (ASTRÖM; MURRAY, 2010).

6 Results and Discussion

This chapter presents the results of the sensitivity analysis tests. It begins with an analysis of the frequency response, specifically of the sensitivity functions plots for the architectures after the optimization procedure performed in Chapter 4. It follows a discussion of the outputs from the Monte Carlo experiments performed according to the conditions described in Section 5.2.1.

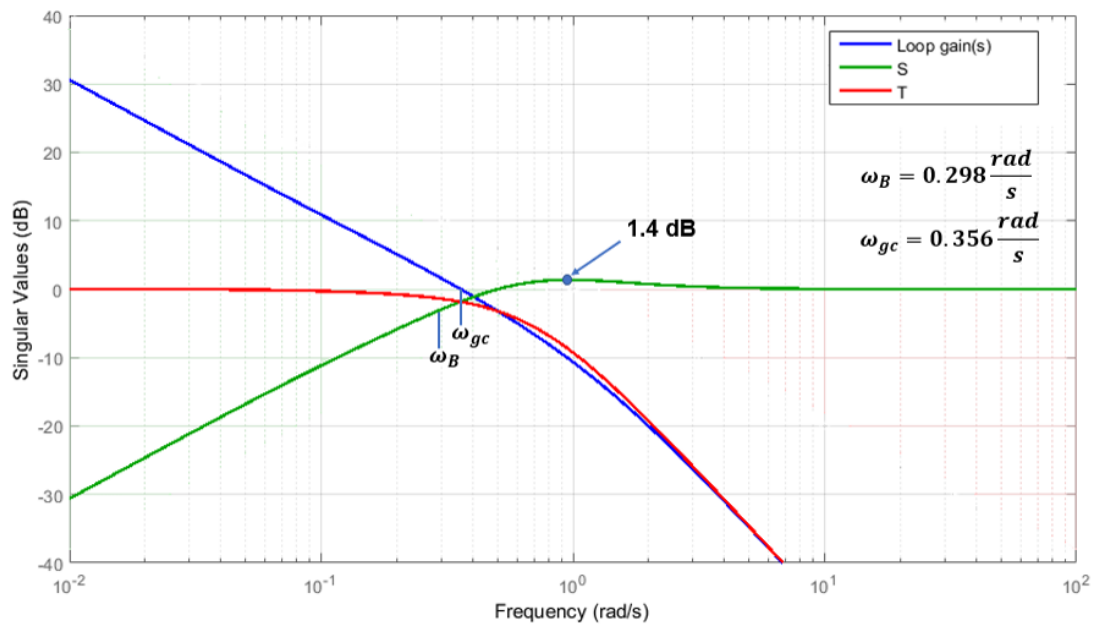


Figure 41 – Loop shape for the longitudinal dynamics of the current architecture after optimization.

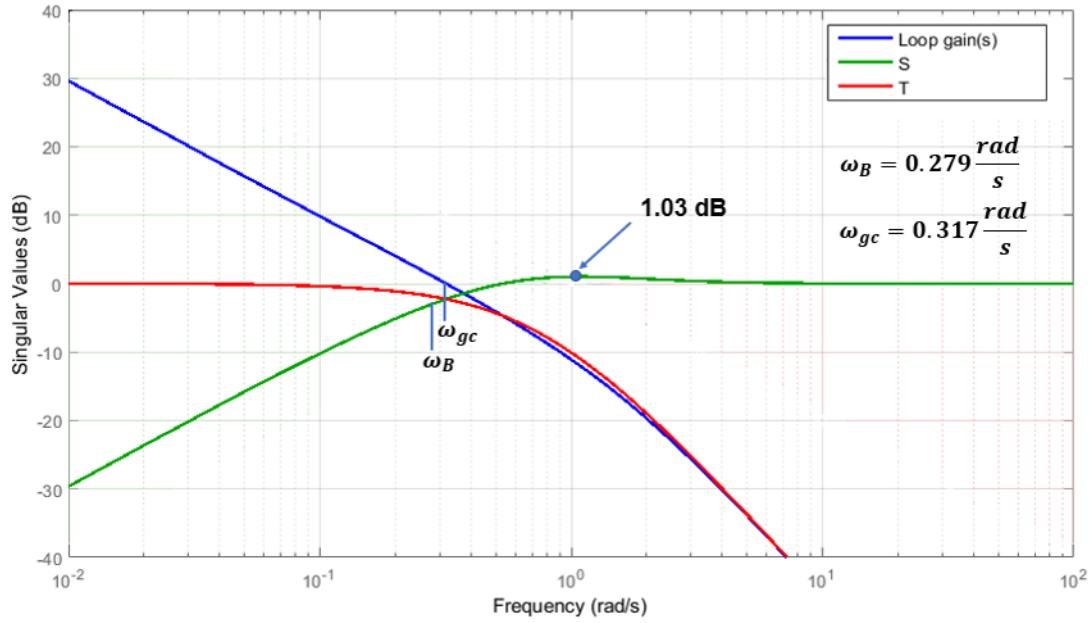


Figure 42 – Loop shape for the longitudinal dynamics of the proposed architecture after optimization.

As previously mentioned in Section 4.4.1, in order to evaluate the performance of the optimization method for gain tuning, the stability margins were verified before and after the optimization for the longitudinal dynamics. As can be seen in Table 9, the margins decreased after the optimization, as expected, since the system was made faster by the tuning procedure. A faster (high bandwidth) system presents less stability margins.

Table 9 – Stability margins before and after optimization for the longitudinal dynamics, where GM and PM stands for the Gain and Phase Margins, respectively.

	Margins	Before	After
<i>Current Architecture</i>	GM	16.2	14.6
	PM	72.4	68.8
<i>Proposed Architecture</i>	GM	19.1	17
	PM	77.3	73.9

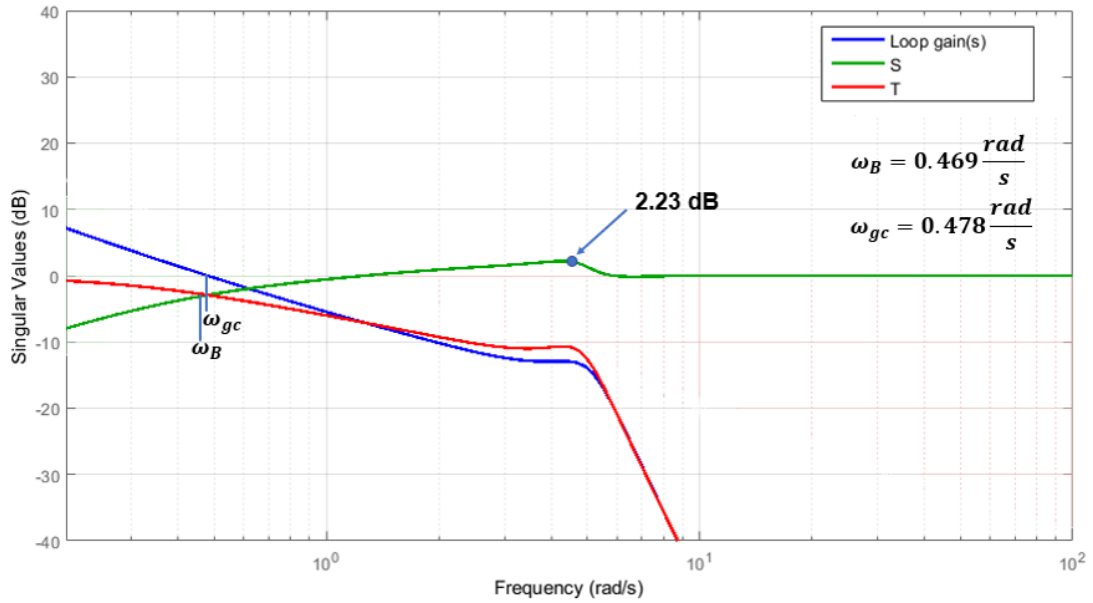


Figure 43 – Loop shape for the lateral dynamics of the current architecture after optimization.

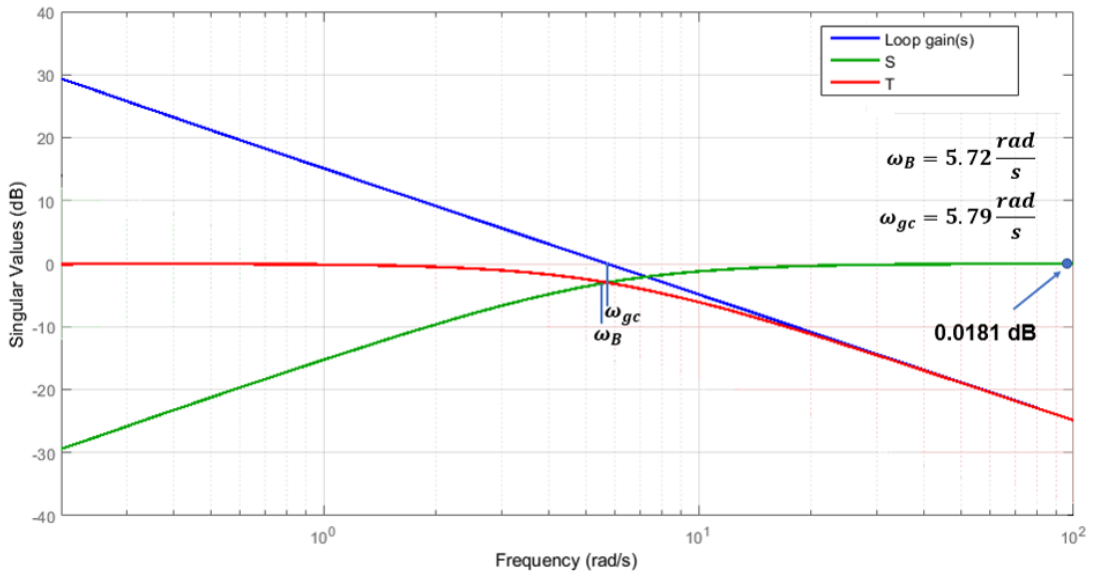


Figure 44 – Loop shape for the lateral dynamics of the proposed architecture after optimization.

As can be seen in Figures 41 to 44, the final loop shape of the control architectures clearly shows the characteristics of the closed loop system. The parameters of interest are the frequencies, ω_{gc} and ω_B , which indicate the system's speed of response. Also, there is the sensitivity peak, M_s , which is a robustness indicator. A value up to 6dB is acceptable for the sensitivity peak (SKOGESTAD; POSTLETHWAITE, 2007), although a lower peak value is recommended. This is directly related to the ability of the system to reject disturbances, and it confirms the previous results. Since the sensitivity peak was higher for the current architecture, it presented more overshoot in the response to disturbances.

The longitudinal dynamics are shown in Figures 41 and 42, for the current and proposed architectures, respectively. As observed, the responses are close in terms of speed, but the sensitivity peak is lower for the proposed architecture.

Figures 43 and 44 show the frequency response for the lateral dynamics. The bandwidth of the current architecture is 0.469rad/s , while for the proposed architecture the value is 5.72rad/s , a factor more than 10 times larger. With respect to the sensitivity function, the current architecture has a value of 2.23dB whereas for the proposed architecture it is only 0.0181dB . This means that the proposed scheme is faster and more robust in terms of low frequency disturbances, but it is more sensitive to measurement noise. The previous control scheme presents a worse performance for disturbance rejection since the sensitivity peak is higher, but it shows good attenuation of high frequency noise.

As stated in Section 5.3, a high bandwidth makes the system vulnerable to noisy measurements, as more high frequency signals enter the system. The good capacity for noise attenuation of the current architecture is shown by the high frequency roll-off (reduction of attenuation) in Figure 43. This is not true for the proposed architecture, as illustrated in Figure 44, the roll-off is much less steep.

In principle, the conclusions obtained from the frequency response analysis should be in line with the Monte Carlo simulations outputs which will follow in the next paragraphs. The advantage of analysing the problem from different perspectives is the possibility of drawing general conclusions and recommendations.

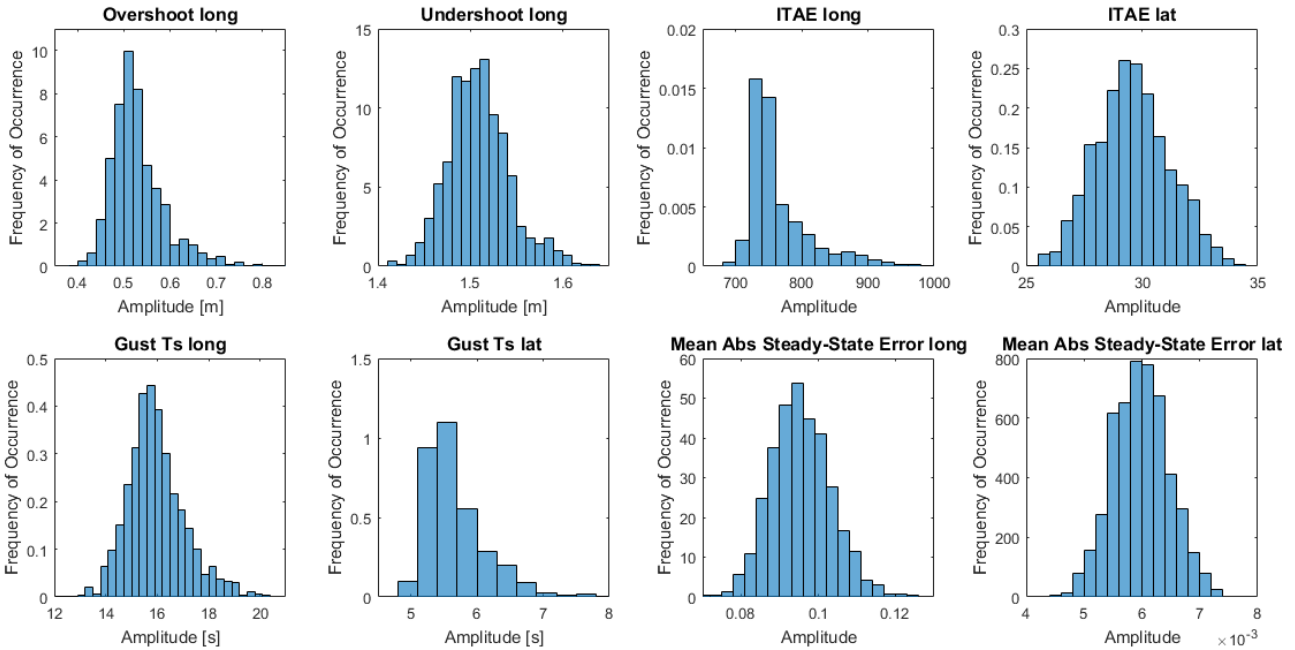


Figure 45 – Resulting distributions from the Monte Carlo simulations of the current architecture with linear models, without time delay.

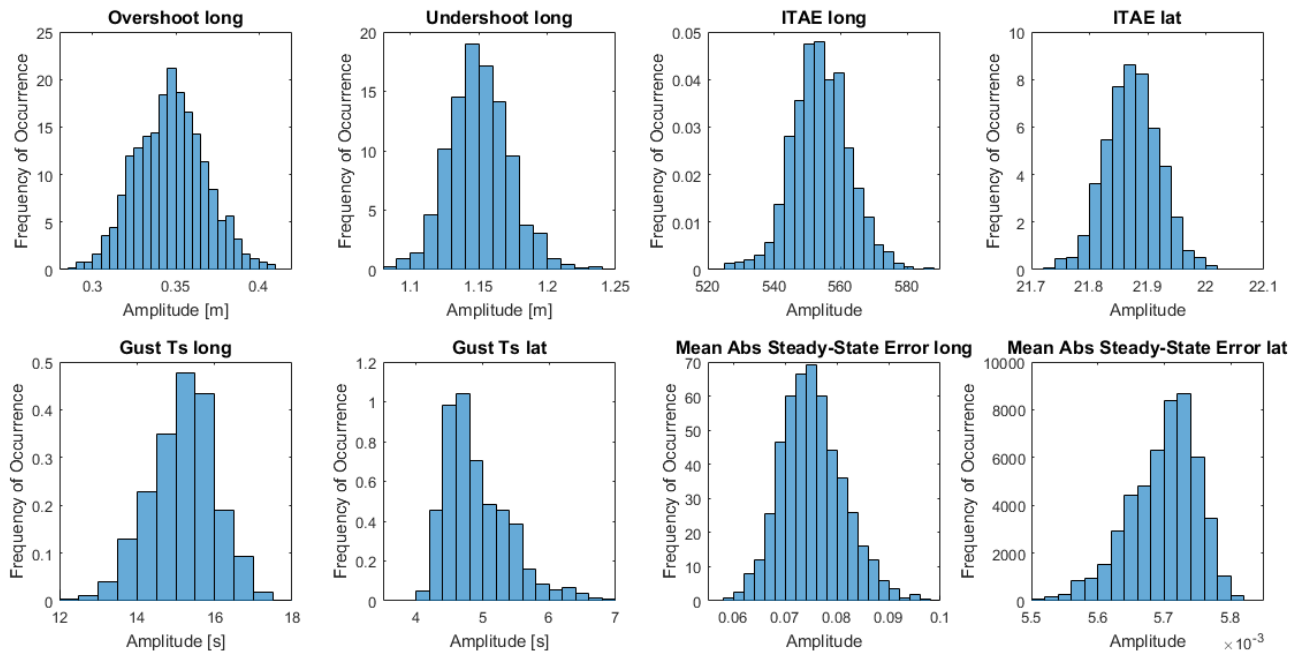


Figure 46 – Resulting distributions from the Monte Carlo simulations of the proposed architecture with linear models, without time delay.

Comparing the plots in Figures 45 and 46 it is possible to see that the proposed architecture presents a better overall performance. In the following paragraphs these differences are highlighted.

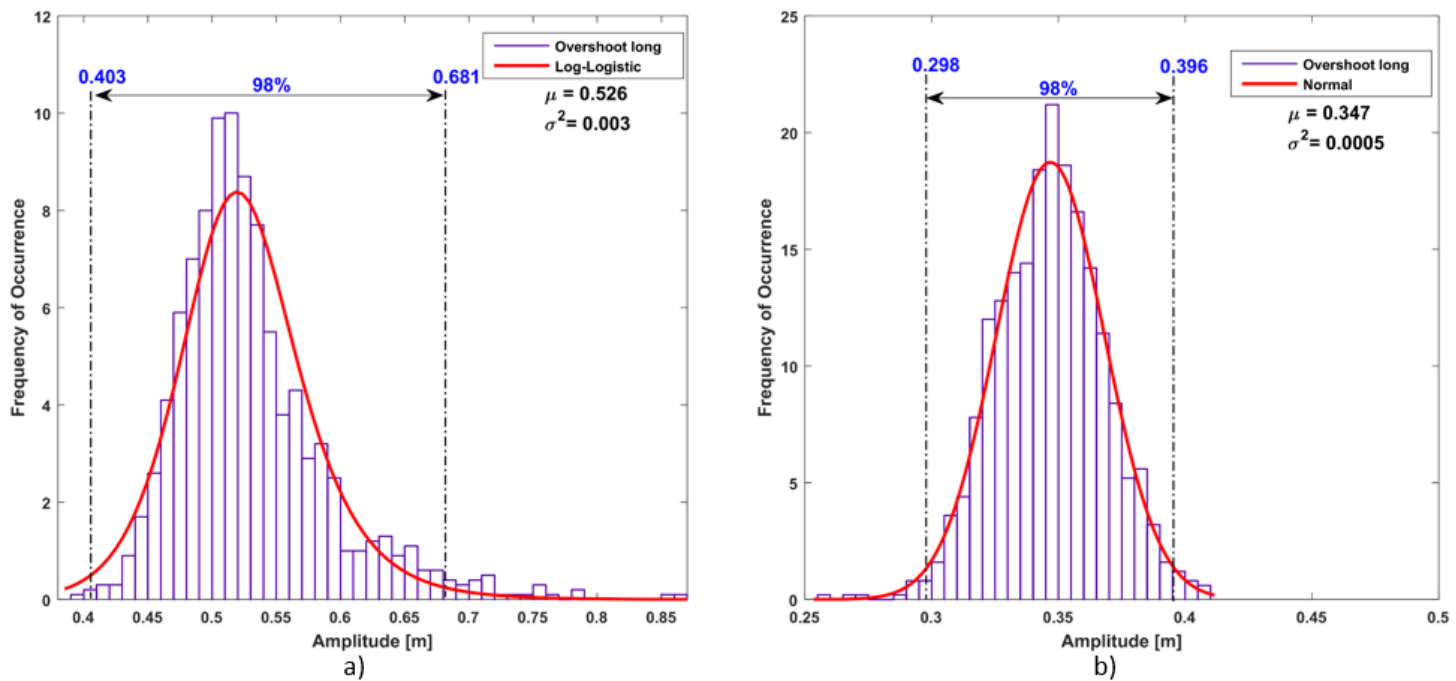


Figure 47 – Overshoot for both control architectures, where 98% is the interval of confidence considered, μ is the mean value and ρ is the variance. a) Current architecture and b) proposed architecture.

Figure 47 shows the maximum overshoot for each architecture. It can be seen that the new architecture can better attenuate disturbances. In a) the mean value for the overshoot is $0.526m$, with a variance of 0.003 , which means that for a 98% confidence interval the values will be between $0.403m$ and $0.681m$. While in b) the mean is $0.347m$ with a variance of 0.0005 , thus it can be said with a confidence of 98% that the overshoot will be between $0.298m$ to $0.396m$. The confidence interval is a range of values so defined that there is a specified probability that the value of a parameter lies within it. A similar behavior has been found for the undershoot, as illustrated in Figure 48, the proposed architecture provides a better attenuation of disturbances.

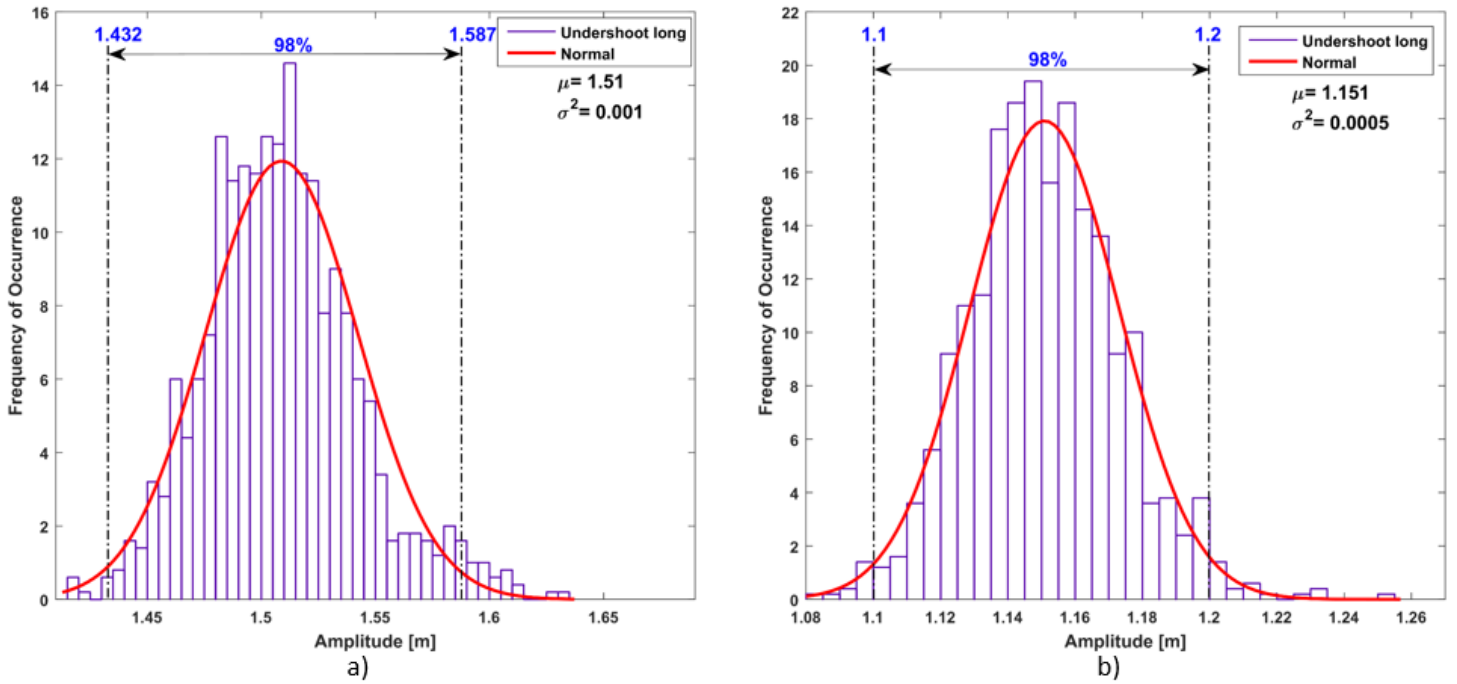


Figure 48 – Undershoot for a) current architecture and b) proposed architecture.

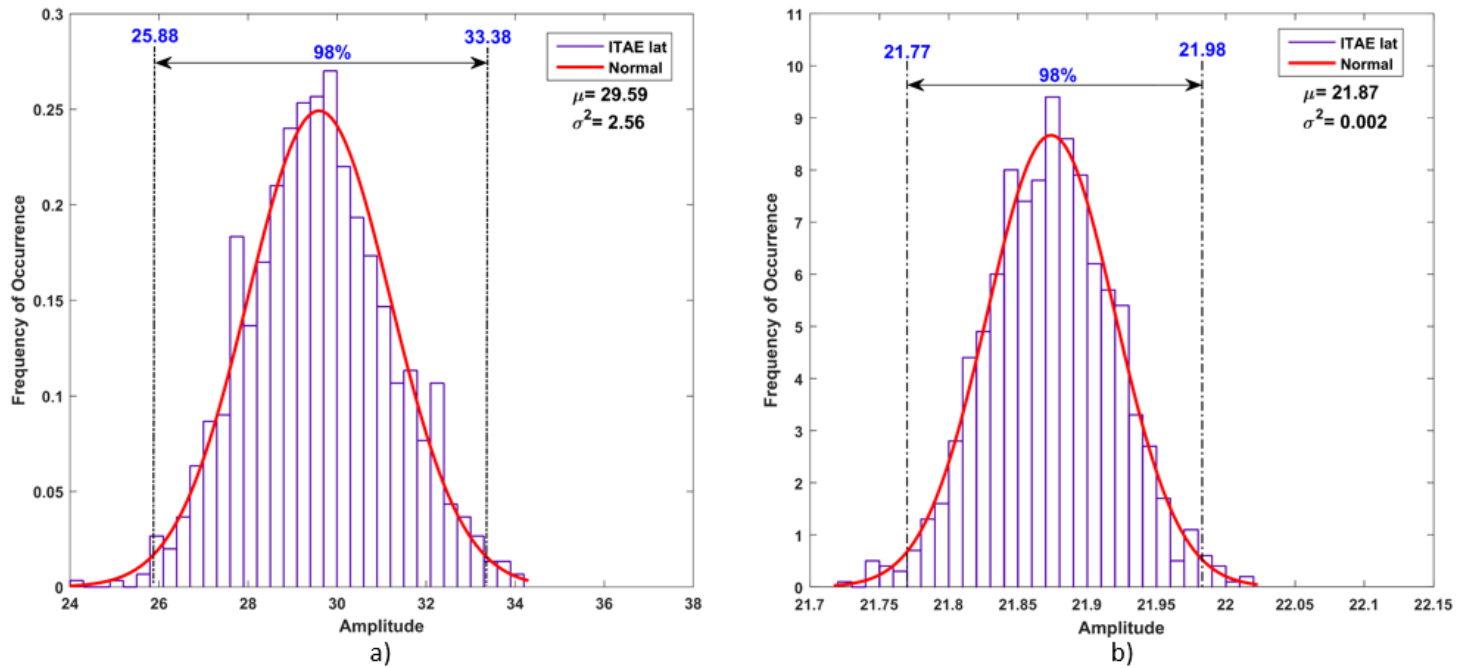


Figure 49 – ITAE for a) current architecture and b) proposed architecture.

Figure 49 shows the ITAE criteria for both architectures. As previously stated, this performance index gives extra emphasis to the error at a later instant of time, after the transitory has died out; it concentrates on the disturbance contribution to the final response. In a) the current architecture presents a mean of 29.6 and a variance of 2.56. For the proposed architecture, the mean is 21.87, with a variance of 0.002. This means the current architecture has a worse performance in terms of disturbance rejection.

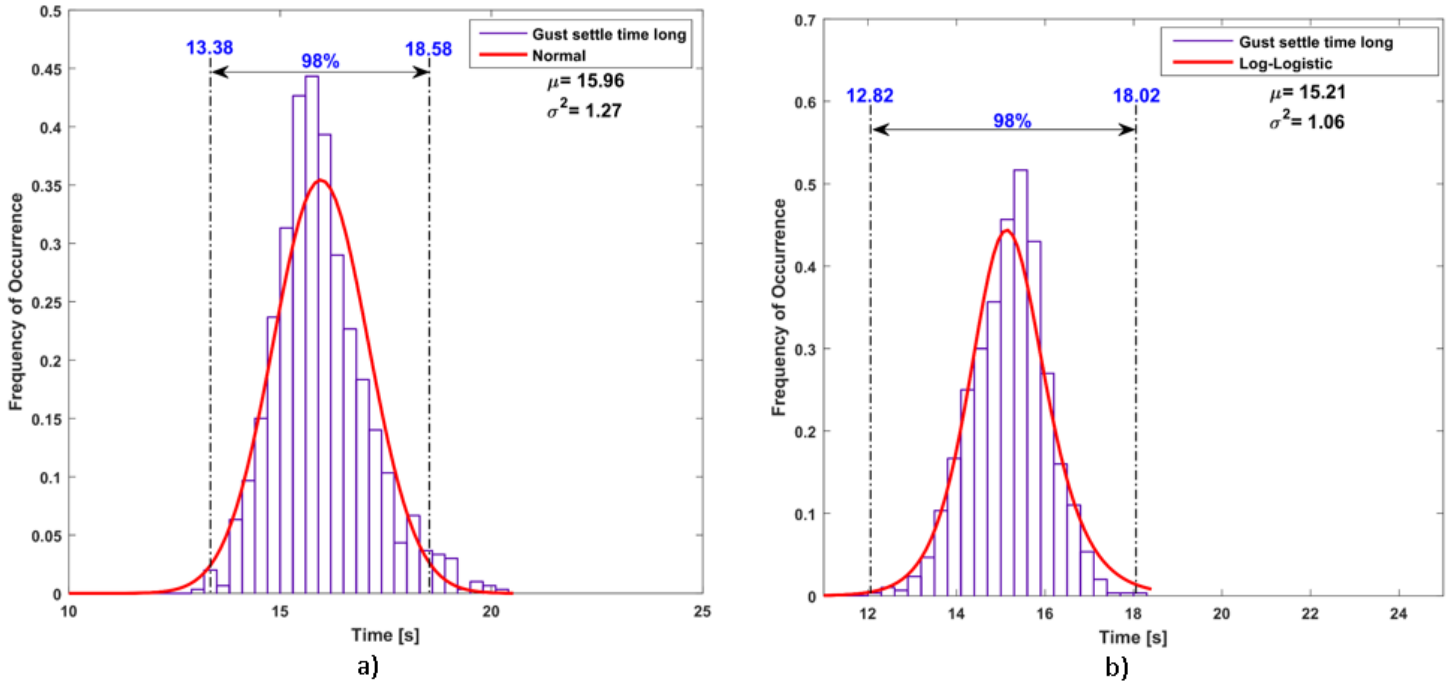


Figure 50 – Time to settle after a gust in the longitudinal direction for a) current architecture and b) proposed architecture.

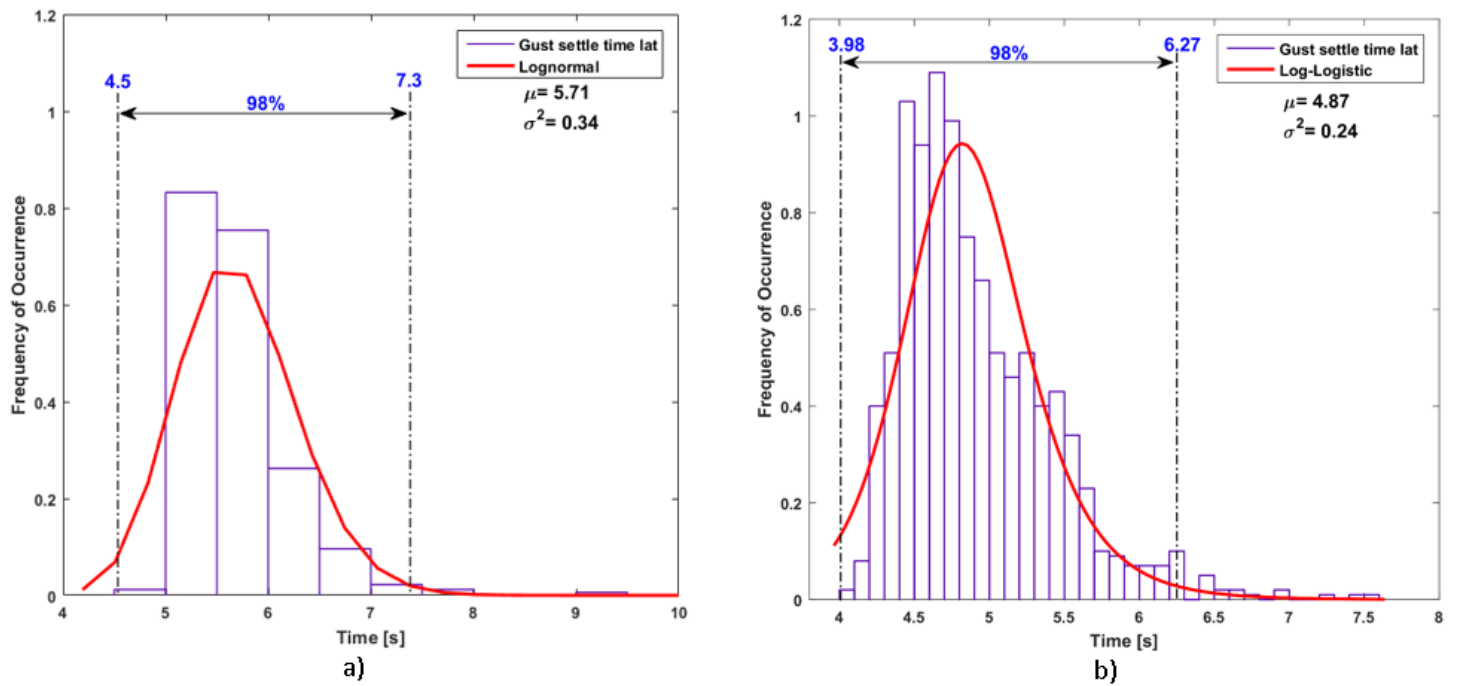


Figure 51 – Time to settle after a gust in the lateral direction for a) current architecture and b) proposed architecture.

Figures 50 and 51 illustrate the gust settling time for the current and proposed architectures, in both directions. It provides a measure of speed of response to disturbances. The proposed scheme seems to be somewhat faster. This is intuitive, since in this structure both

vehicles correct for differences in position. Thus, the system should present a shorter reaction time. Although there is not a large difference from the mean value found in the corresponding distributions from this experiments, it is important to remember that the actual landing is carried out with a relatively high speed. Therefore, any second counts. Also, these results corroborate the ones found in Section in 4.6.

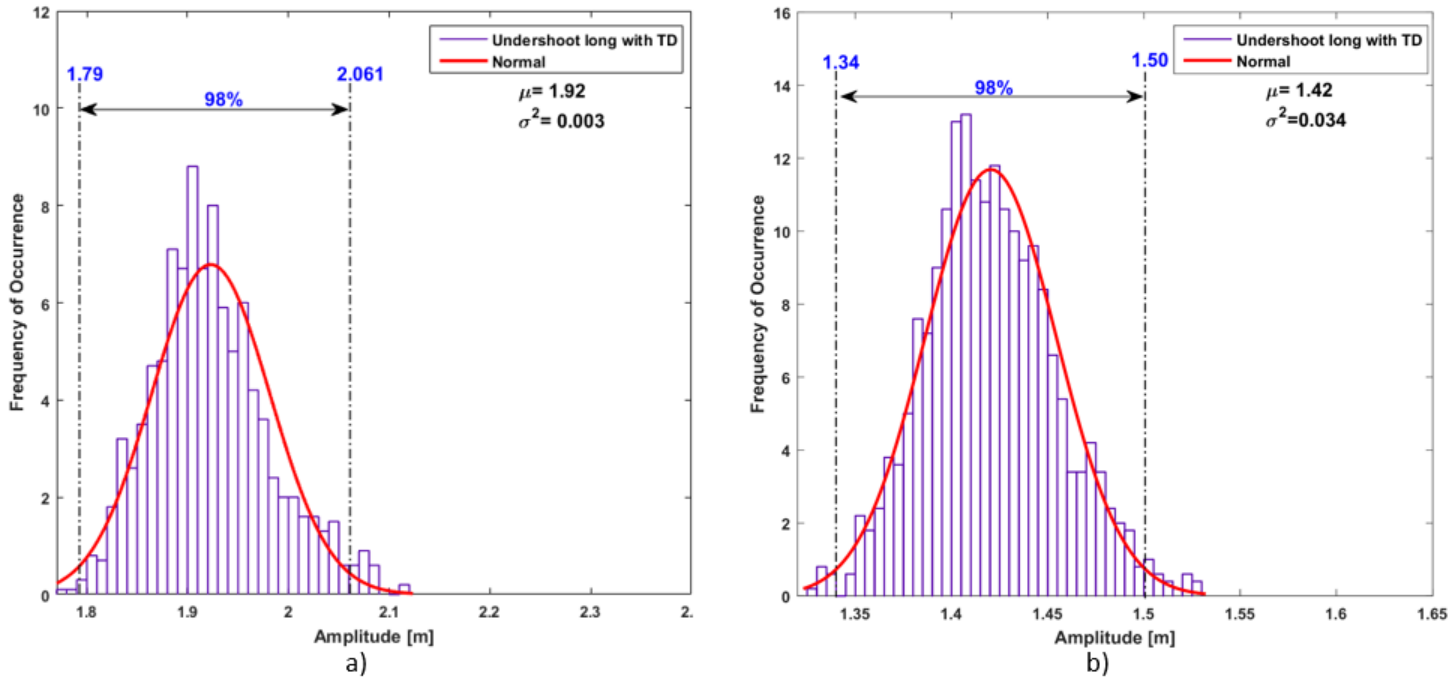


Figure 52 – Undershoot in the presence of time delay for a) current architecture and b) proposed architecture.

Figure 52 shows the results in the presence of time delay where it is visible its contribution to the final response. The time delay increased the undershoot in the longitudinal direction for both architectures. In fact, it undermines the response of the overall system. In the proposed architecture, both vehicles receive each other positions and correct for position errors, so more information is exchanged in the network. As a result, it is expected to be more affected by communication lags. However, as stated before, this delay should be smaller for the next flight experiments and safety measures (see Section 4.5) have been adopted to avoid instability. Hence, the time delay should not play a major role in the upcoming real experiments.

Although these results show the new architecture has some advantages, it is important to emphasise that the previous design is simpler to implement and it works well under normal conditions. Thus, the reader should evaluate which structure would be more appropriate depending on the expected external conditions. For example, on a day with strong crosswinds, the proposed architecture would be more appropriated. Also, when a shorter landing time is required, the new architecture seems more suitable since it is faster in the presence of disturbances. However, if the measurement data of a given aircraft are excessively noisy even after filtering and the gain in speed is not so relevant, it could be better to use the current control scheme.

7 Conclusion and Outlook

The main contribution of this thesis has been to fill in the parts not covered in previous works, mainly a thorough comparison between control architectures and the development of systematic methods for controller tuning and analysis of control performance. These objectives were achieved and the methods presented can be used, with adjustments, for future developments within the project. Certainly it would be worth doing additional Monte Carlo experiments with the complete nonlinear simulation, as it allows to obtain other variables of interest such as the number of successful landings, the amount of runway used and the time until touchdown, just to name a few. It could also possibly be employed in the verification of the autoland controller for large solar aircrafts which is currently under development in the group. Surely, there is still room for improvement within the cooperative landing architecture, mostly in the controller design, as a more sophisticated strategy could be used in which the robustness aspect is considered already prior to the design of the controller. This would make the design less dependent of a specific model, as it would work assuming certain variables and inputs to the model will be unknown but bounded. H-infinity loop shaping has promising characteristics and it is vastly used for robust control design (SKOGESTAD; POSTLETHWAITE, 2007) and could be an alternative.

The work in the Flying Robots Group will continue and the next step will be to test the proposed modifications in the cooperative architecture as described in Chapter 3 in actual landing experiments. Additionally, new landing experiments will be performed in combination with an MPC controller (PAVANI, 2018), which might then be interfaced in a cascaded configuration with the proposed control architecture.

Bibliography

- ÅSTRÖM, K. J.; HÄGGLUND, T. *PID controllers: theory, design, and tuning*. 2. ed. [S.l.]: Instrument society of America Research Triangle Park, NC, 1995. 1-2, 164-166 p. Cited in pages 52, 55, and 98.
- ASTRÖM, K. J.; MURRAY, R. M. *Feedback systems: an introduction for scientists and engineers*. [S.l.]: Princeton university press, 2010. 65–68, 319–321 p. Cited in pages 28, 47, 48, 55, 62, 72, 74, 101, 102, and 104.
- BALMER, G. R. *Modelling and Control of a Fixed-wing UAV for Landings on a Mobile Landing Platform*. Dissertation (Master Thesis) — German Aerospace Center (DLR), Robotics and Mechatronics Center and KTH Royal Institute of Technology, 2015. Cited in pages 22, 27, 28, 66, 92, and 93.
- BEARD, R. W.; MCLAIN, T. W. *Small unmanned aircraft: Theory and practice*. [S.l.]: Princeton university press, 2012. 95-97 p. Cited in pages 29, 89, 90, and 93.
- BLAKELOCK, J. H. *Automatic control of aircraft and missiles*. 2. ed. [S.l.]: John Wiley & Sons, 1991. 17-18 p. Cited in pages 37 and 89.
- CHALK, C. R. et al. *Background Information and User Guide for Mil-F-8785B (ASG), 'Military Specification-Flying Qualities of Piloted Airplanes'*. [S.l.], 1969. Cited in page 48.
- CLOSE, C. M.; NEWELL, J. C.; FREDERICK, D. K. *Modeling and analysis of dynamic systems*. 3. ed. [S.l.]: John Wiley & Sons, 2002. 299-316 p. Cited in page 29.
- COOK, M. V. *Flight dynamics principles: a linear systems approach to aircraft stability and control*. 3rd. ed. [S.l.]: Butterworth-Heinemann, 2012. Cited in page 29.
- EC-SAFEMOBIL. *EC-SAFEMOBIL*. 2016. <<http://ec-safemobil-project.eu/>>. [Online; accessed 13-May-2018]. Cited in pages 44, 45, and 46.
- FRANKLIN, G. F. et al. *Feedback control of dynamic systems*. [S.l.]: Prentice Hall, 2009. 100-103 p. Cited in pages 37 and 96.
- GOLNARAGHI, F.; KUO, B. *Automatic control systems*. 9. ed. [S.l.]: John Wiley & Sons, 2010. 278-279 p. Cited in page 96.
- HAROLD, K.; RANDAL, A. *Simulation of dynamical system with MATLAB and Simulink*. [S.l.: s.n.], 2011. 576–590 p. Cited in pages 65, 67, 69, 97, and 98.
- KONG, J. et al. Kinematic and dynamic vehicle models for autonomous driving control design. In: IEEE. *Intelligent Vehicles Symposium (IV), 2015 IEEE*. [S.l.], 2015. p. 1094–1099. Cited in pages 28 and 104.
- LAGARIAS, J. C. et al. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on optimization*, SIAM, v. 9, n. 1, p. 112–147, 1998. Cited in page 97.

LAIACKER, M. et al. Vision aided automatic landing system for fixed wing uav. In: IEEE. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Tokyo, Japan, 2013. p. 2971–2976. Cited in page 22.

LANGELAAN, J. W.; ALLEY, N.; NEIDHOEFER, J. Wind field estimation for small unmanned aerial vehicles. *Journal of Guidance, Control, and Dynamics*, v. 34, n. 4, p. 1016–1030, 2011. Cited in page 49.

LAVRETSKY, E. F/a-18 autonomous formation flight control system design. *AIAA paper*, v. 4757, 2002. Cited in page 38.

LI, Y.; ANG, K. H.; CHONG, G. C. Pid control system analysis and design. *IEEE Control Systems*, IEEE, v. 26, n. 1, p. 32–41, 2006. Cited in pages 38 and 96.

MASON, S. J. *On the logic of feedback*. Thesis (PhD) — Massachusetts Institute of Technology, 1952. Cited in page 47.

MATLAB. *version 7.10.0 (R2018b)*. Natick, Massachusetts: The MathWorks Inc., 2018. Cited in page 49.

MURRAY, R. M. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, American Society of Mechanical Engineers, v. 129, n. 5, p. 571–583, 2007. Cited in page 25.

MUSKARDIN, T. et al. A novel landing system to increase payload capacity and operational availability of high altitude long endurance uav. In: IEEE. *2016 IEEE International Conference on Unmanned Aircraft Systems (ICUAS)*. Arlington, USA, 2016. p. 495–504. Cited in pages 22, 36, and 44.

MUSKARDIN, T. et al. A novel landing system to increase payload capacity and operational availability of high altitude long endurance uavs. *Journal of Intelligent & Robotic Systems*, Springer, v. 88, n. 2-4, p. 597–618, 2017. Cited in pages 22, 24, 26, 36, and 44.

MUSKARDIN, T. et al. Landing of a fixed-wing uav on a mobile ground vehicle. In: IEEE. *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden, 2016. p. 1237–1242. Cited in page 22.

NELSON, R. C. et al. *Flight stability and automatic control*. [S.l.]: WCB/McGraw Hill New York, 1998. Cited in page 30.

PAVANI, M. F. F. *Implementation and Analysis of a Model Predictive Controller for Landing Fixed-Wing Aircraft on Mobile Platforms*. [S.l.], 2018. Available at: <<https://elib.dlr.de/120003/>>. Cited in pages 38 and 84.

PERSSE, L. *Cooperative Control for Landing a Fixed-Wing Unmanned Aerial Vehicle on a Ground Vehicle*. Dissertation (Master Thesis) — German Aerospace Center (DLR), Robotics and Mechatronics Center and KTH Royal Institute of Technology, 2016. Cited in pages 22, 30, 31, and 42.

RYU, J.-H.; ARTIGAS, J.; PREUSCHE, C. A passive bilateral control scheme for a teleoperator with time-varying communication delay. *Mechatronics*, Elsevier, v. 20, n. 7, p. 812–823, 2010. Cited in page 61.

SALTELLI, A. et al. *Global sensitivity analysis: the primer*. [S.l.]: John Wiley & Sons, 2008. Cited in page 67.

SKOGESTAD, S.; POSTLETHWAITE, I. *Multivariable feedback control: analysis and design*. [S.l.]: Wiley New York, 2007. Cited in pages 50, 51, 67, 77, and 84.

STEIN, G. Respect the unstable. *IEEE Control Systems*, IEEE, v. 23, n. 4, p. 12–25, 2003. Cited in pages 73 and 74.

UAV FACTORY. *Penguin BE Electric Unmanned Platform*: Datasheet. Irvington, NY, USA. Cited in pages 25 and 110.

UNMANNED DYNAMICS. *AeroSim – aeronautical simulation blockset*: User's guide. Hood River, OR, USA. Cited in page 27.

Appendix

APPENDIX A – Individual Vehicle Control Design

Contains details about the control design for the individual vehicles and explains the optimization method employed for the tuning of the controller parameters.

When considered from a higher level perspective, both vehicles (UAV and UGV) are assumed to respond to commands in a deterministic way, such that they can be treated as black boxes. The inputs are desired velocity and course angle while the outputs are positions and velocities. Therefore, before addressing the cooperative control problem, the controllers for the linear vehicle models derived in Chapter 2 has to be designed.

A.1 Flight Control

In order to stabilize and make the aircraft follow commands, a control scheme has to be designed. The primary goal of an autopilot system is to control position, velocity and attitude. For the autopilot design, a simplification is adopted where the longitudinal dynamics are considered decoupled from the lateral dynamics (BLAKELOCK, 1991). This assumption facilitates the control design without considerably affecting the model accuracy. The procedure used for the control design is described in the following subsections.

A.1.1 Successive Loop Closure

The classical method of choice for controlling aircraft consists on designing cascaded Single-Input-Single-Output (SISO) control loops, where the loops are closed successively. The innermost loop is commonly referred to as stability augmentation, which is basically a feedback of the angular rates. Then, it is important to make sure the following loop is 5 to 10 times faster than the previous one (BEARD; MCLAIN, 2012). This speed measure is referred to as bandwidth in the frequency domain. The higher the bandwidth, the faster the system. This procedure is illustrated in Figures 53 and 54 where the innerloop is modeled as a unity gain due to the bandwidth separation.

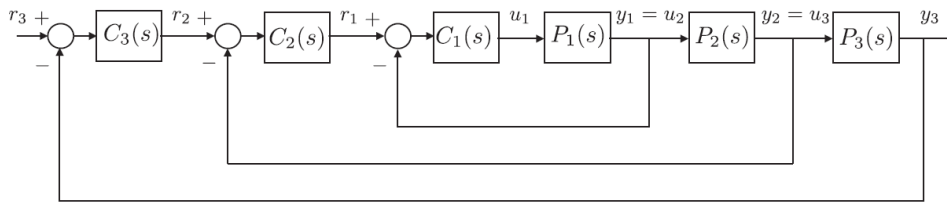


Figure 53 – Successive loop closure design.

Source: (BEARD; MCLAIN, 2012).

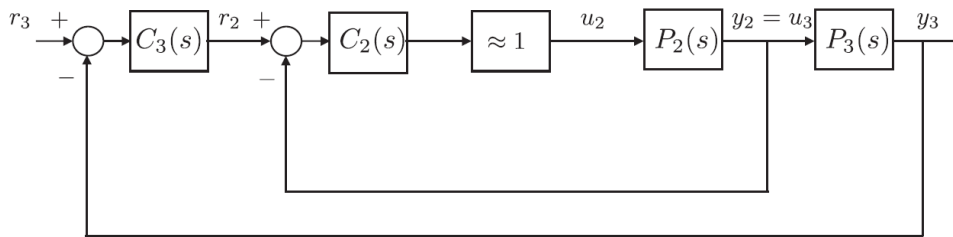


Figure 54 – Successive loop closure design with inner loop modeled as a unity gain.

Source: (BEARD; MCLAIN, 2012).

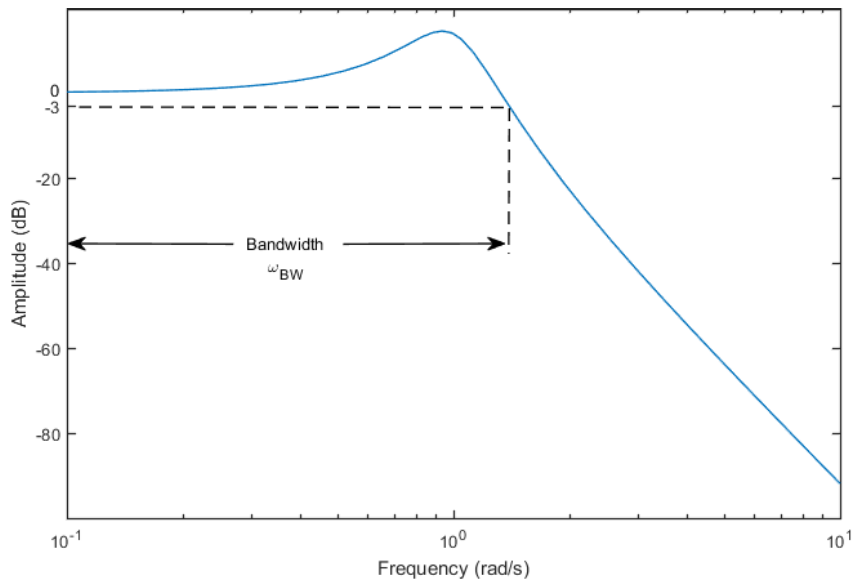


Figure 55 – Bandwidth as seen from a Bode plot.

As illustrated in Figure 55, one of the definitions of bandwidth is the frequency at which the closed-loop system response crosses the $-3dB$ line on a bode plot. Additional frequency-domain measures will be further explained in Section 4.3.

A.1.2 Lateral Control

In this subsection the strategies for the lateral control of the aircraft are explained.

The Penguin BE has a vertical stabilizer which generates restoring forces that helps to stabilize the movement around the yaw axis. During flight experiments, it was observed that a sufficient turning performance could be obtained with the aileron only, and thus no rudder control was added for the time being.

A.1.2.1 Lateral Stability Augmentation and Autopilot

As can be seen in Figure 56, the lateral controller consists of three cascaded SISO loops. The design starts with the innermost loop, which is a proportional controller with feedback of the roll rate p . It essentially acts as a damping term for the attitude ϕ controller.

$$\delta_a = K_p (p_{des} - p) \quad (A.1)$$

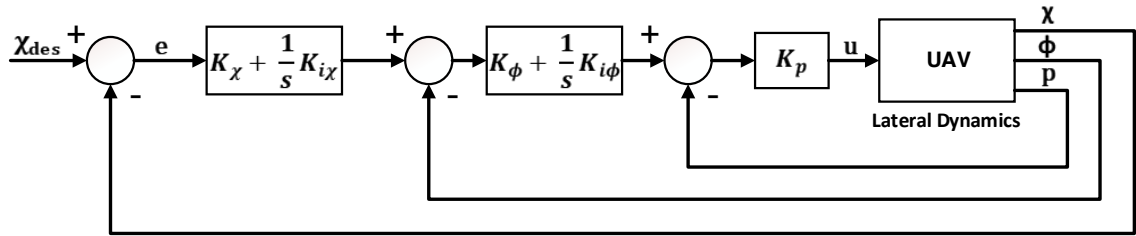


Figure 56 – Lateral Autopilot.

The bank angle ϕ is controlled by a PI-controller, described by the equation below

$$p_{des} = (K_\phi + \frac{1}{s} K_{i\phi}) \cdot (\phi_{des} - \phi) \quad (A.2)$$

The outermost loop controls the course χ and it is referred to as an autopilot. It was implemented as a PI controller with feedback of the course angle, which is measured with reference to the north. Its output is a course rate $\dot{\chi}_{des}$. However, as the following inner loop described in Equation A.2 expects a desired bank angle ϕ_{des} , it is necessary to calculate it according to Equation A.4 and illustrated in Figure 57.

$$\dot{\chi}_{des} = (K_\chi + \frac{1}{s} K_{i\chi}) (\chi_{des} - \chi) \quad (A.3)$$

$$\phi_{des} = \arctan \left(\frac{V_k \dot{\chi}_{des}}{g} \right) \quad (A.4)$$

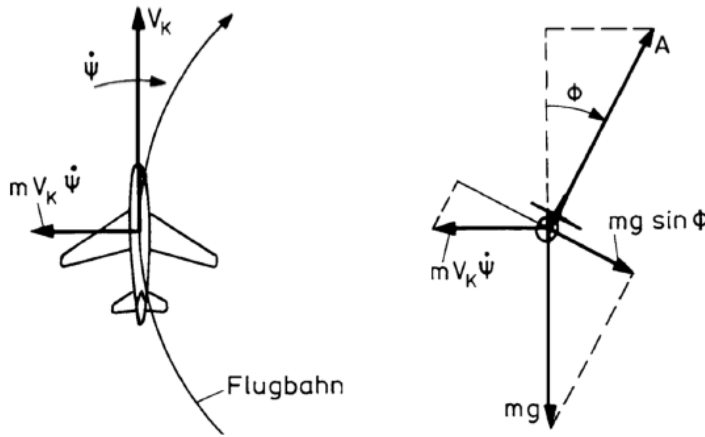


Figure 57 – Force equilibrium in a steady turn. The lift force A is tilted by the angle ϕ . Its vertical component $A \cos(\phi)$ is in equilibrium with the weight while its horizontal component is in equilibrium with the centrifugal force $m V_k \dot{\psi}$. Since $\dot{\psi} \approx \dot{\chi}$ when there is no wind, the relation $\tan(\phi) = \frac{V_k \dot{\chi}}{g}$ holds.

Source: (BALMER, 2015).

From the transfer functions derived from model 2.2, it was verified that the inner loop consisting of the feedback of the roll rate in series with the PI controller for the roll angle had a bandwidth of 6.47 rad/s , while the outer loop for the course angle presented a bandwidth of 1.18 rad/s , a factor 5.48 smaller, thus guaranteeing the recommended bandwidth separation.

A.1.3 Longitudinal Control

In this subsection the strategies for the longitudinal control of the aircraft are explained.

A.1.3.1 Longitudinal Stability Augmentation and Autopilot

Firstly, the underdamped short-period mode was handled by adding a pitch rate feedback. Then, the inner loop for pitch angle was added. The final configuration consisting of a PI controller for θ and the pitch rate damper is illustrated in Figure 58. The resulting controller is described by:

$$\delta_e = -\left(K_\theta + \frac{K_{i\theta}}{s}\right)(\theta_{des} - \theta) - K_q q \quad (\text{A.5})$$

and the thrust loop was also closed by a PI controller with a feedforward term. This structure is represented in Figure 59 and described by the equation below:

$$\delta_t = -\left(K_\tau + \frac{K_{i\tau}}{s}\right)(\tau_{des} - \tau) + \delta_{tFF} \quad (\text{A.6})$$

Usually the longitudinal control is done in a similar manner as the one described previously for the lateral case, closing successive loops considering the required bandwidth separation.

The problem with this approach is that there is an inherent coupling between velocity and altitude commands in the longitudinal motion. For instance, when a climb is commanded, the altitude controller gives a nose-up command through the elevator, and this causes the aircraft to slow down. Due to the decrease in airspeed, the lift force also decreases and the airplane starts to lose altitude. Also, when a step command is given in throttle, the aircraft short term reaction is to increase its airspeed and altitude. This coupling effect is usually not a desirable behavior, and for the cooperative landing it can not be tolerated, since it is required an independent control of forward speed and sink rate as is further explained in Chapter 3.

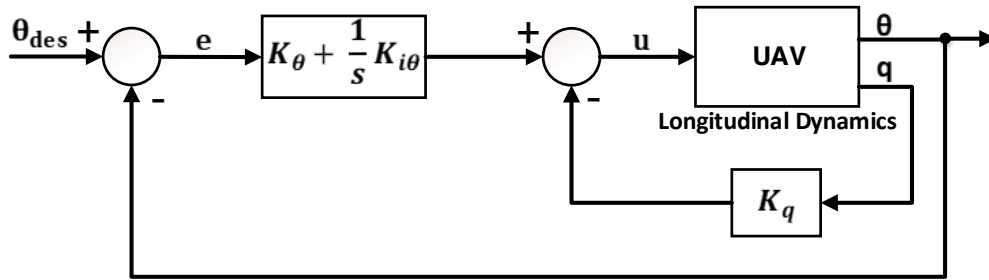


Figure 58 – Pitch angle inner loop.

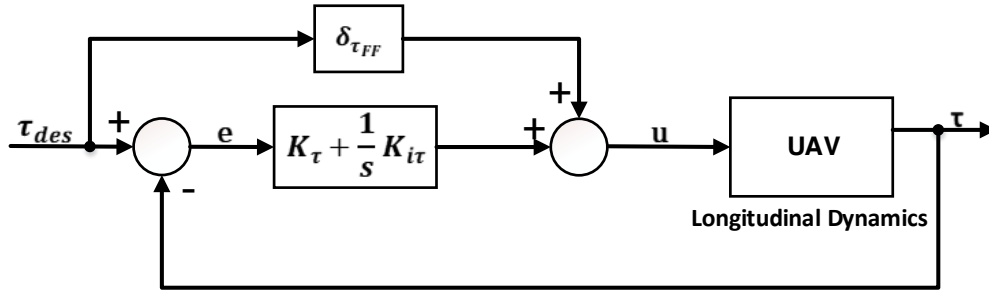


Figure 59 – Thrust inner loop.

A.1.4 TECS

The independent control of altitude and airspeed can be troublesome to achieve using classical methods only, due to its strong coupling. A way to reduce this coupling effect is to operate the system under different flight regimes (BEARD; MCLAIN, 2012), which generally results in a non-optimal performance. An alternate solution is to make use of a method based on energy distribution; it distributes the total energy of the system between its kinematic and potential forms. This method is called Total Energy Control System (TECS), in this controller structure the throttle is responsible for adding energy into the system while the elevator distributes this energy accordingly. A detailed description of the implementation of TECS for the Penguin BE can be found in Balmer (2015).

The complete diagram of the aircraft longitudinal autopilot is shown in Figure 60.

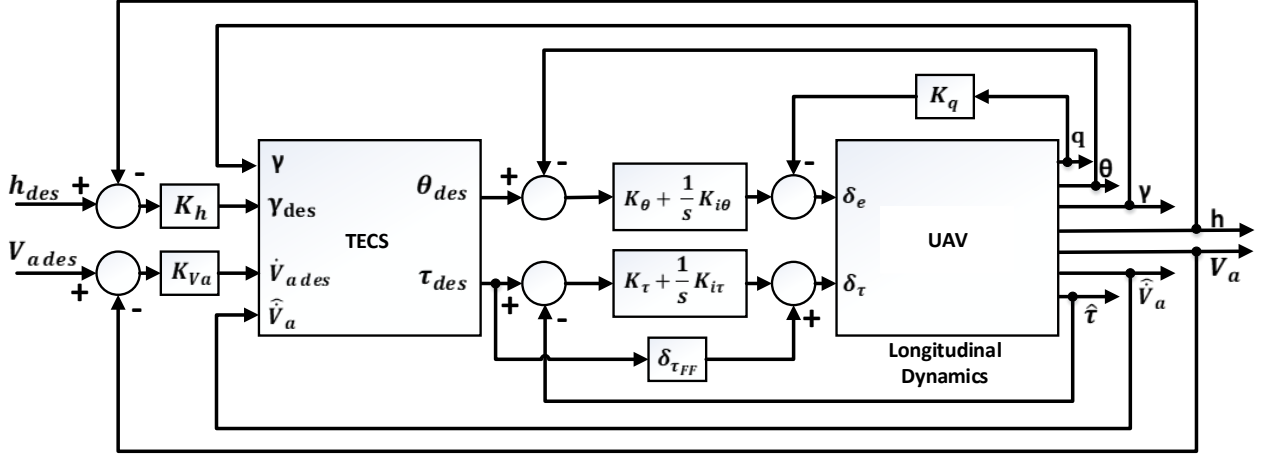


Figure 60 – Longitudinal Autopilot.

A.2 Ground vehicle control

In this section the longitudinal and lateral control strategies for the ground vehicle are explained.

A.2.1 Longitudinal Control

The simple PI-controller scheme of Figure 61 is enough to achieve a good performance for the car velocity. The control design method of choice was pole placement. Usually for the pole placement technique the system is approximated as a second order system. Equation A.7 provides its representation in canonical form. Using the time-domain specifications of Table 10 and the second order relations of Table 11, it is possible to place the closed-loop poles in specific locations in the s -plane as shown in Figure 62, in order to achieve the desired response. Although the plant is of higher order as shown in Section 2.3.2, using pole placement considering it as a second order system will give a good initial guess for the gain values. Then, an optimization algorithm can be used to find an optimal set of gains, according to time-domain requirements. This algorithm basically calculates some time-domain measures from the output signal of the system and changes the controller gains accordingly, in order to alter the behavior of the system and thus satisfying the given requirements; this method will be further explained in Section A.2.3.

Second order canonical form:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (\text{A.7})$$

where ω_n is the natural frequency and ζ is the damping ratio of the system. The way this values are calculated can be seen in Figure 62.

$$s = -\zeta\omega_n \pm \omega_n\sqrt{1 - \zeta^2} \quad (\text{A.8})$$

From the especifications of Table 10 and Equation A.8 the desired closed-loop poles are defined as:

$$s = -0.213 \pm 0.29j$$

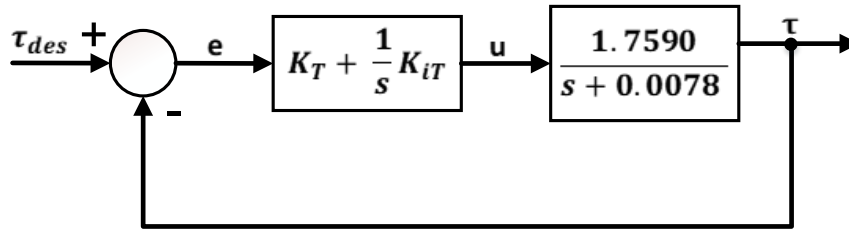


Figure 61 – Car cruise control.

The transfer function in Figure 61 was obtained from the state space model of the car 2.9, and it is the relation between the throttle input f and the velocity in the forward direction \dot{x} . Although the original system is of fifth order, it can be reduced to a first order system due to pole-zero cancellations.

Table 10 – Time-domain specifications.

Specifications
Rise time < 5s
Overshoot < 10%
Steady State error < 2%

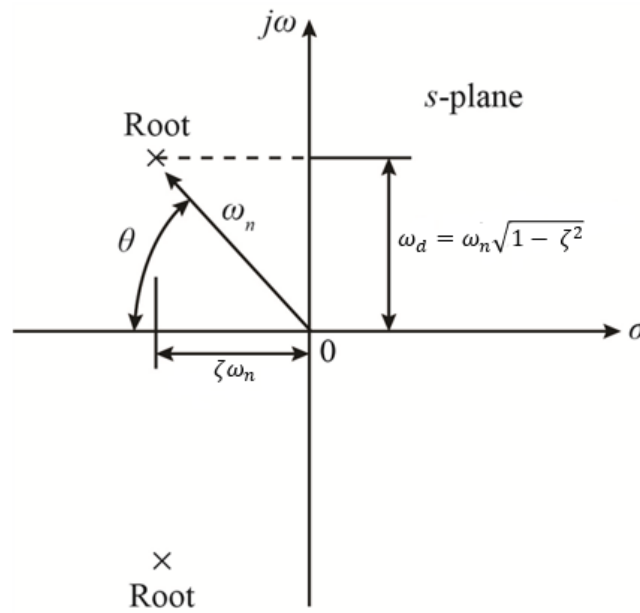


Figure 62 – The s-plane, where ω_d is the damped frequency, ω_n is the natural frequency, and ζ is the damping ratio.

Source: Adapted from (GOLNARAGHI; KUO, 2010).

Table 11 – Second order relations

Property	Formula
Settling Time (2 %)	$t_s = \frac{4}{\zeta\omega_n}$
Rise Time	$t_r = \frac{1.8}{\omega_n}$
Peak Time	$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}}$
Overshoot	$OS_{max} = e^{-\zeta\pi/\sqrt{1-\zeta^2}}$
Damping Ratio	$\zeta = \frac{-\ln(M_p)}{\sqrt{\pi^2 + \ln^2(M_p)}}$

Source: (FRANKLIN et al., 2009).

Table 12 – Effects of independent P, I and D tuning on closed-loop response. For example, while K_I and K_D are held fixed, increasing K_P can decrease rise time, increase overshoot, slightly increase settling time, decrease the steady-state error, and decrease stability margins.

	Rise Time	Overshoot	Settling Time	Steady State Error
$K_p(\uparrow)$	Decrease	Increase	Small Change	Decrease
$K_i(\uparrow)$	Decrease	Increase	Increase	Decrease
$K_d(\uparrow)$	Small Change	Decrease	Decrease	No Change

Source: (LI; ANG; CHONG, 2006).

Thus, considering the second order canonical form in Equation A.7, the coefficients of the PI controller are matched to the coefficients of the characteristic equation of the desired closed-loop system, in order to find the following values:

$$K_T = 0.2376$$

$$K_{iT} = 0.0737$$

A.2.2 Lateral control

For the lateral control a simple proportional controller was sufficient, since the lateral dynamics already has free integrators, the steady-state error is zero and it was not necessary to add an integral term. In order to find the initial guess, the *sisotool* interface from MATLAB® has been used. Considering the requisites of Table 10, the following value was found:

$$K_\chi = 0.5$$

A.2.3 Optimization Method

In the process of control design, many times the values for the controller parameters are chosen based on manual tuning and the designer's experience. Alternatively, as shown in the previous section, they can be determined through a low order approximation. However, in general such methods lead to suboptimal solutions. To make this process more efficient and systematic, it was used an optimization method to select the controller parameters which minimize a performance index.

A multiparameter objective function was specified to evaluate the performance of the closed-loop system and iteratively run a Simulink® model changing the controller parameters at each iteration (HAROLD; RANDAL, 2011).

A variation of the MATLAB® built-in function *fminsearch* from the optimization toolbox was used. This modification is called *fminsearchbnd*, in which it is possible to specify lower and upper bounds for the parameters, and is available on the MATLAB® file exchange page. This function uses a nonlinear optimization algorithm; from an initial estimate, it finds the minimum of a multivariable function without calculating the gradient. The function uses the Nelder-Mead simplex algorithm, as described in Lagarias et al. (1998). This algorithm tries to minimize a nonlinear function of n real variables, using only the function value, without differentiation.

Considering the UGV cruise control illustrated in Figure 61 as an example, the control parameters K_T and K_{iT} must be selected to optimize the system's response to a step input in commanded speed. There are numerous measures that can be used to evaluate the system's step

response. Some of them, as suggested by Harold and Randal (2011) and Åström and Hägglund (1995), are defined as follows

- **Rise time** (t_r) \rightarrow Time required for the response to go from 10 to 90% of its final value.
- **Maximum overshoot** (OS_{max}) \rightarrow Difference between maximum and final value in underdamped systems.
- **Maximum derivative of control signal** ($|\dot{u}|$) \rightarrow Maximum rate of change of control command, in the car cruise control example is the throttle derivative.
- **Integral Squared Error** (ISE) \rightarrow Integral of squared error from time zero to infinity.
- **Integral Absolute Error** (IAE) \rightarrow Integral of absolute value of error from zero to infinity.
- **Integral Time-weighted Absolute Error** (ITAE) \rightarrow Integral of absolute value of error multiplied by the time, from zero to infinity.

The objective function f is assumed to be a function of these measures, that is,

$$f(t_r, OS_{max}, |\dot{u}|, ISE, IAE, ITAE) = F(K_T, K_{iT}) \quad (\text{A.9})$$

The goal is to find optimum values for the gains, so that

$$f_{opt} = \min_{(K_T > 0, K_{iT} > 0)} F(K_T, K_{iT}) \quad (\text{A.10})$$

In this problem, f is defined as a linear combination of the following performance measures

$$f(t_r, OS_{max}, |\dot{u}|, ITAE) = w_1 t_r + w_2 OS_{max} + w_3 |\dot{u}| + w_4 ITAE \quad (\text{A.11})$$

Note that the objective function is implicitly dependent on K_T and K_{iT} because each measure t_r , OS_{max} , \dot{u} , ISE , IAE , and $ITAE$ depends on these parameters.

Different combinations of performance measures were tested and the form of Equation A.11 was considered more appropriate for this application.

The selection of weights w_1 , w_2 , w_3 e w_4 may vary according to the design goal. When all terms are equally relevant to the final response, it is common to choose them in a way that all terms contribute equally to the final cost. For example, if the rise time term outputs a value of 3, while the ITAE term outputs a value of 300, then the weight for the rise time should be 100.

The same method has been used to find the proportional gain for the UGV lateral controller. Moreover, it has also been employed to tune the gains of the UAV lateral controller and in the inner loops of the longitudinal autopilot. The TECS gains were not tuned by this method, as the values from the last successful landing experiments were kept unchanged. For this tuning method it is only required to give the starting point as well as the lower and upper bounds of the variables to the optimization algorithm.

A.2.4 Optimal set of gains

Using the method described above, optimal set of gains were found for the vehicles controllers, as shown in Tables 13 and 14.

Table 13 shows the values of the UAV controller structures illustrated in Figures 56, 58, 59 and 60. The values in Table 14 correspond to the UGV controllers described in Section A.2.1.

Table 13 – Values tuned with the optimization method for the UAV linear model.

Gain	Optimal value
K_θ	-5.734
$K_{i\theta}$	-1.001
K_q	0.9478
K_h	0.01628
K_τ	4.24
$K_{i\tau}$	0.784
δ_{FF}	1
K_{Va}	0.06022
K_ϕ	7.435
$K_{i\phi}$	0.1
K_p	-0.4456
K_χ	2.0864
K_{TP}	1
K_{TI}	2
K_{EP}	1
K_{EI}	2

Table 14 – Values tuned with the optimization method for the UGV linear model.

Gain	Optimal value
K_T	0.7894
K_{iT}	0.1084
K_χ	0.895

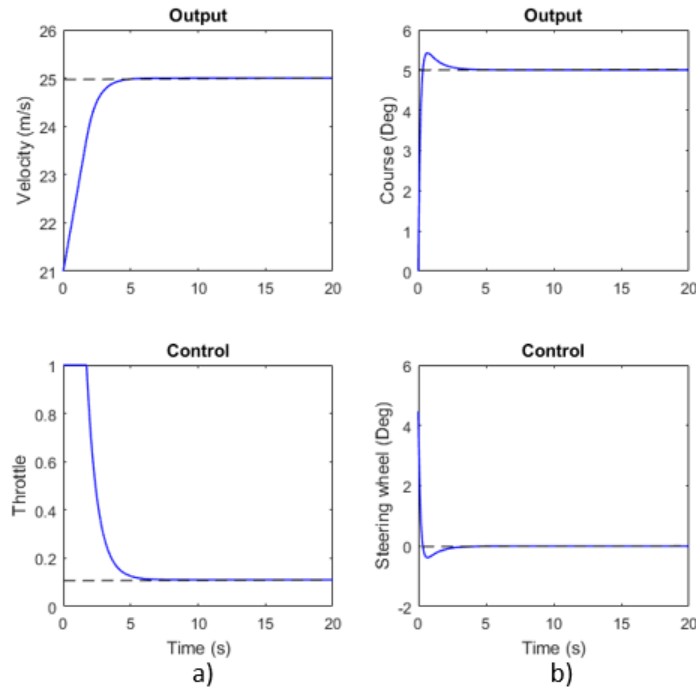


Figure 63 – UGV response to steps in a) velocity and b) course angle.

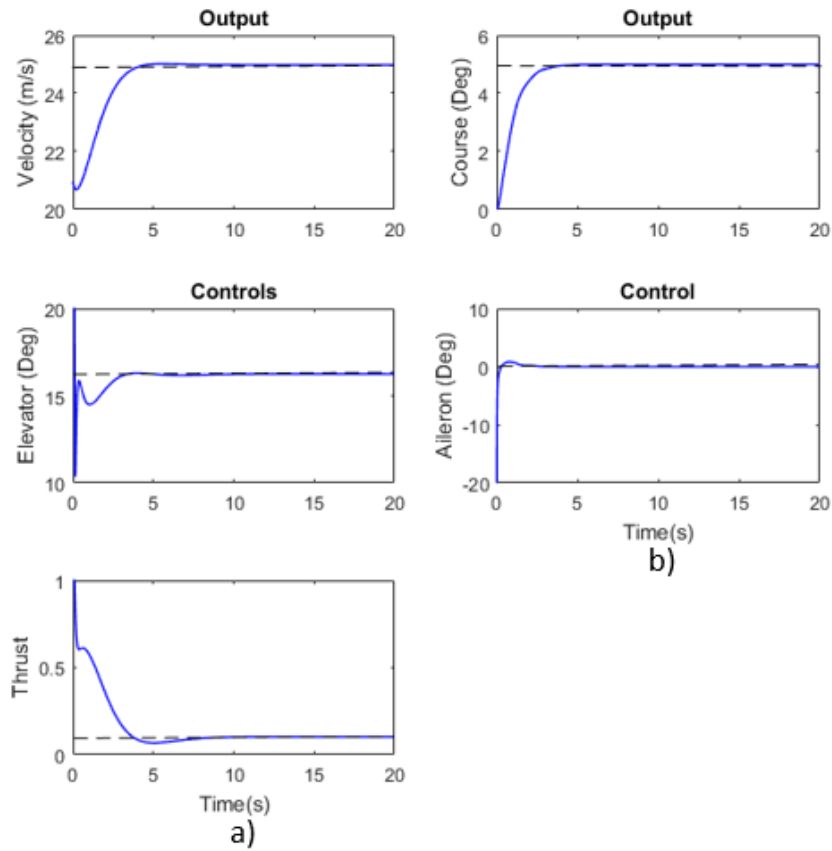


Figure 64 – UAV response to steps in a) velocity and b) course angle.

Figures 63 and 64 show the closed-loop step responses for the UGV and UAV, both vehicles satisfied the specifications in Table 10.

APPENDIX B – UGV Modeling

B.1 Cruise

To develop the mathematical model for the car, according to Aström and Murray (2010), the first step is to write down the force balance equation:

$$m \frac{dv}{du} = F_{eng} + F_{aero} + F_{frict} + F_g \quad (B.1)$$

where v is the speed of the car, m is the total mass, F_{eng} is the force generated by the engine, F_{aero} , F_{frict} and F_g are the forces due to aerodynamic drag, rolling friction and gravity, respectively, and can be define as:

$$F_{aero} = -\frac{1}{2}\rho C_d A v |v| \quad (B.2)$$

$$F_{frict} = -mg C_r \text{sgn}(v) \quad (B.3)$$

$$F_g = -mg \sin(\theta) \quad (B.4)$$

where

$g \Rightarrow$ is the gravitational constant, $g = 9.8 \text{ m/s}^2$

$C_r \Rightarrow$ is the coefficient of rolling friction, a typical value is $C_r = 0.01$

$\text{sgn}(v) \Rightarrow$ is the sign of v

$\rho \Rightarrow$ is the density of the air, a typical value is $\rho = 1.3 \text{ k/m}^3$

$C_d \Rightarrow$ is the aerodynamic drag coefficient, a typical value is $C_r = 0.32$

$A \Rightarrow$ is the front area of the car

The force generated by the engine is directly proportional to its torque, which is proportional to the rate of fuel injection, which is itself proportional to a control signal $0 \leq u \leq 1$ that controls the throttle position. The torque also depends on engine speed ω . The torque at full throttle is given by:

$$T(\omega) = T_m \left(1 - \beta \left(\frac{\omega}{\omega_m} - 1 \right)^2 \right) \quad (B.5)$$

where the maximum torque T_m is obtained at the maximum engine speed ω_m . For the Audi A6 Avant 2.5 TDI, these parameters are $T_m = 268.1 Nm$, $\omega_m = 460 rad/s$ and $\beta = 0.4$.

The engine speed is related to the velocity through the equation

$$\omega = \frac{n}{r}v = \alpha_n v \quad (B.6)$$

and the driving force can be written as

$$F = \frac{nu}{r}T(\omega) = \alpha_n u T(\alpha_n v) \quad (B.7)$$

where r is the wheel radius, δ_t is the throttle deflection and n is the transmission ratio. According to the datasheet provided by the manufacturer of the car (referred to Appendix C), the gear ratios for the Audi A6 Avant are $g_1 = 3.665$, $g_2 = 1.999$, $g_3 = 1.407$, $g_4 = 1$, $g_5 = 0.742$

$$n = g_n * (axle\ ratio) \quad (B.8)$$

Figure 65 shows the torque as a function of the vehicle speed. Basically, the effect of the gear shift is to flatten the torque curve so that a full torque can be obtained over the whole speed range.

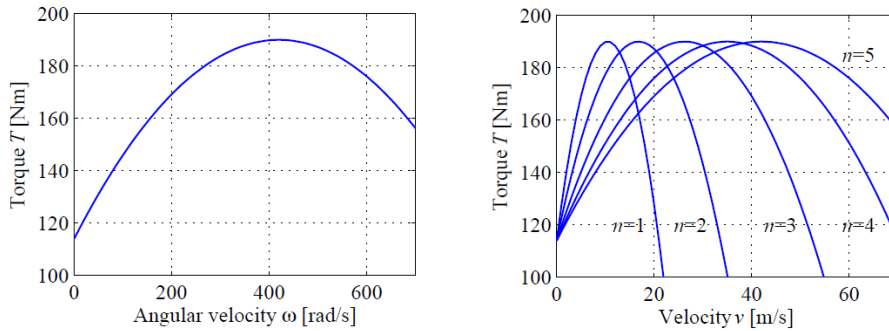


Figure 65 – Torque curve.

Source: (ASTRÖM; MURRAY, 2010).

$$\frac{\partial f_1}{\partial v} = \frac{\bar{u}\alpha_n T'(\alpha_n \bar{v}) - \rho C A \bar{v}(v - v_e)}{m} \quad (B.9)$$

Steps to derive $T'(\omega)$:

$$T(\omega) = T_m \left(1 - \beta \left(\frac{w}{w_m} - 1 \right)^2 \right)$$

$$T(\omega) = T_m - T_m \beta \left(\frac{\omega}{\omega_m} - 1 \right)^2$$

$$T(\omega) = T_m - T_m \beta \left[\left(\frac{\alpha_n v}{\omega_m} - 1 \right) \left(\frac{\alpha_n v}{\omega_m} - 1 \right) \right]$$

$$T(\omega) = T_m - T_m \beta \left[\frac{\alpha_n^2 v^2}{\omega_m^2} - \frac{\alpha_n v}{\omega_m} - \frac{\alpha_n v}{\omega_m} + 1 \right]$$

$$T(\omega) = T_m - T_m \beta \left[\frac{\alpha_n^2 v^2}{\omega_m^2} - \frac{\alpha_n^2 v}{\omega_m} + 1 \right]$$

$$T'(\omega) = 0 - \frac{2T_m \beta \alpha_n^2 v}{\omega_m^2} - \frac{2T_m \beta \alpha_n}{\omega_m} + 0$$

$$T'(\omega) = \alpha_n \left(\frac{-2T_m \beta \alpha_n v}{\omega_m^2} - \frac{2T_m \beta}{\omega_m} \right)$$

B.2 Steering

In motion control a change in trajectory is accomplished through an actuator that causes a change in the orientation. In an automobile the mechanism responsible for this change in orientation is the steering wheel and in a bicycle is the front wheel. In many cases, it is sufficient to use a simplified model that captures the basic kinematics of the system to understand its behavior.

The nonlinear continuous time equations that describe a kinematic bicycle model in an inertial frame are

$$\dot{x} = v \cos(\psi + \lambda) \tag{B.10}$$

$$\dot{y} = v \sin(\psi + \lambda) \tag{B.11}$$

$$\dot{\psi} = \frac{v}{l_r} \sin(\lambda) \tag{B.12}$$

$$\dot{v} = a \tag{B.13}$$

$$\lambda = \tan^{-1} \left(\frac{l_r}{l_f + l_r} \tan(\delta) \right) \tag{B.14}$$

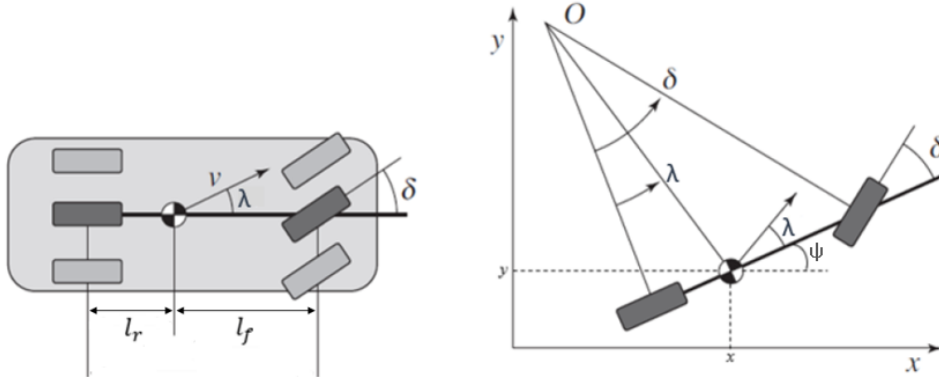


Figure 66 – Vehicle steering dynamics. The left figure shows a top view of a four-wheel vehicle, where l_r is the distance between the center of mass and the rear wheel, and l_f is the distance between the center of mass and the front wheel of the car. By approximating the motion of the front and rear wheels pair by a single front wheel and a single rear wheel, we obtain an simplification called the *bicycle model*.

Source: Adapted from (ASTRÖM; MURRAY, 2010) and (KONG et al., 2015).

where v is the velocity of the center of mass, l_r is the distance between the center of mass and the rear wheel, l_f is the distance between the center of mass and the front wheel of the car, as shown in Figure 66. x and y are the coordinates of the center of mass, ψ is the heading angle and λ is the angle between the velocity vector v and the centerline of the vehicle.

Equations B.10 to B.14 can be used to model a car assuming there is no slip between the wheels and the road and that the two front wheels can be approximated by a single wheel at the center of the car.

B.3 Simulink Blocks Description

In this section the car model as implemented in Simulink® is presented.

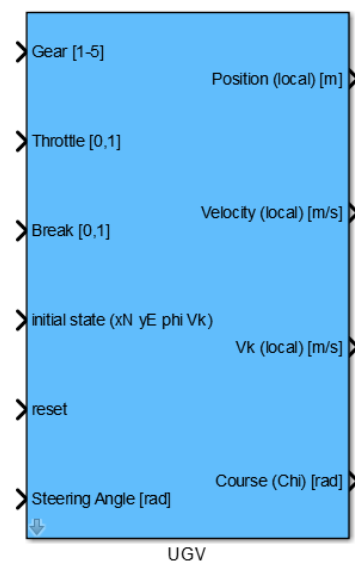


Figure 67 – UGV Simulink mask.

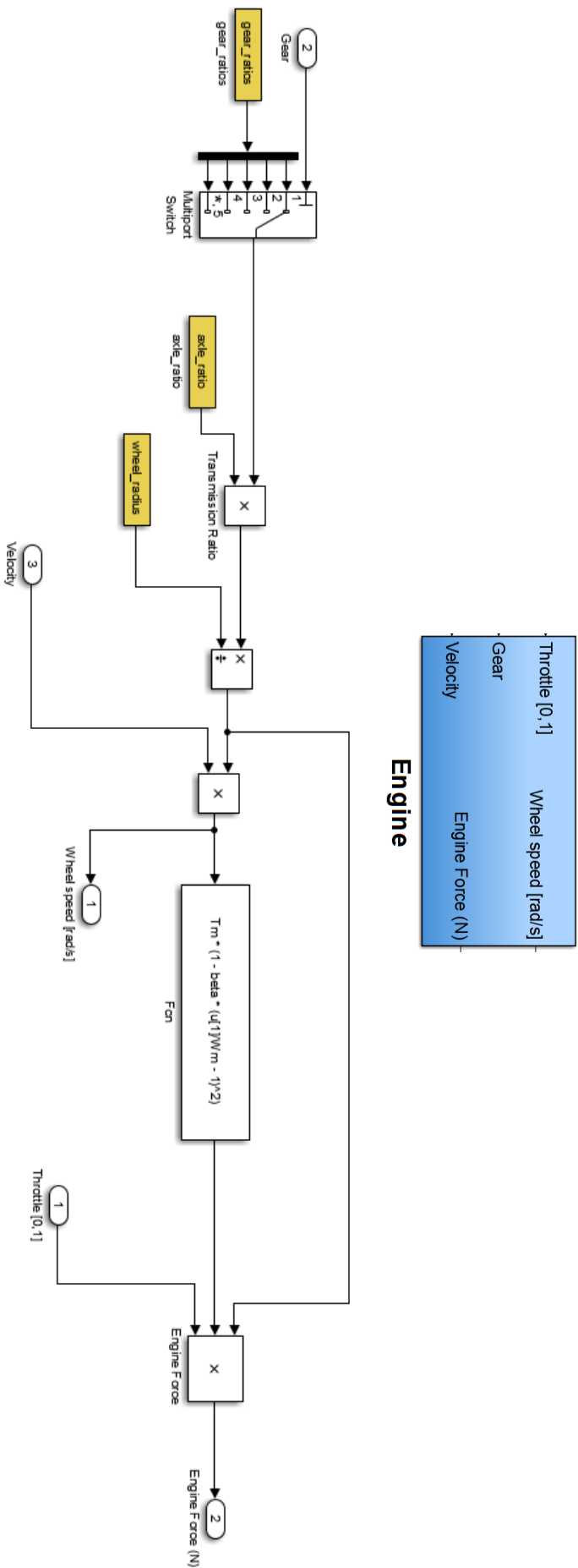


Figure 68 – UGV engine equation.

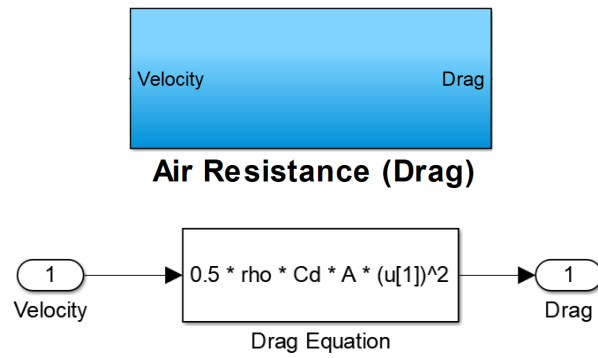


Figure 69 – UGV air resistance equation.

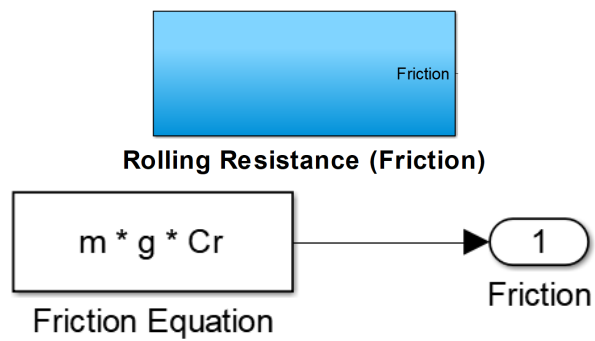


Figure 70 – UGV rolling resistance equation.

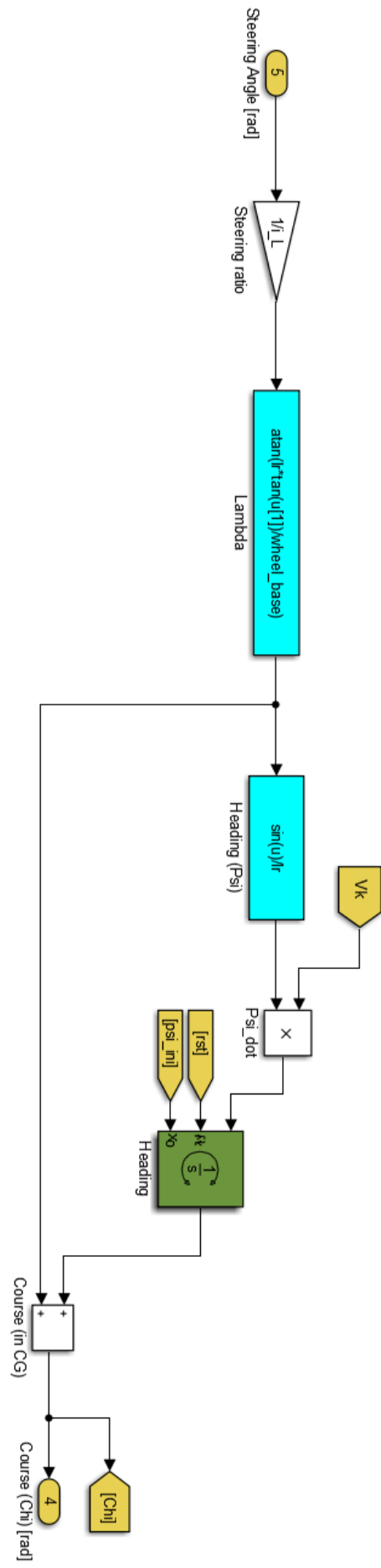


Figure 71 – UGV lateral dynamics.

APPENDIX C – Audi A6 Specifications

The parameters of the car, model Audi A6 Avant 2.5 TDI, year 2000, as previously used in the landing experiments are presented in Table 15.

Table 15 – Audi A6 Specifications.

Parameter	Value
Mass	1900 Kg
Maximum Velocity	219 Km/h
Gear Ratios	[3.665 1.999 1.407 1 0.742]
Axle Ratio	2.984
Wheel Diameter	631.9 mm
l_r	1380 mm
l_f	1380 mm
Wheelbase	2760 mm
Steering Ratio	16.2
Drag Coefficient	0.32
Front Area	2.19 m ²
Air Density	0.32

Source: Audi A6 Datasheet

APPENDIX D – Penguin BE UAV

Specifications

The specifications presented here were taken from the Penguin BE datasheet and are only for reference. The aircraft used in the flight experiments was customized by DLR and thus the values may be slightly different.

Table 16 – Structural Specifications

Specifications	
Empty mass	14.9 kg
Payload capacity	6.6 kg
Maximum takeoff mass	21.5 kg
Wing span	3.3 m
Length	2.27 m
Wing area	0.79 m ²
Propulsion power	electric
Propulsion type	2700 W
Battery type	Li-Po
Battery capacity	640 Wh

Source: UAV Factory.

Table 17 – Performance Specifications

Specifications	
Cruise speed	22 m/s
Stall speed	13 m/s
Max level speed	36 m/s
C_L max (clean wing)	1.3
C_L max (45 flaps)	1.7
Takeoff run	30 m
Endurance	110 min
Ceiling	6000 m

Source: UAV Factory.